FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Vision-Based Feature Matching as a Tool for Robotic Localization

Nolasco Amado dos Santos Napoleão

WORKING VERSION

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Professor Armando Jorge Miranda de Sousa

Co-Supervisor: M.Sc. Válter Joaquim Ramos Costa

July 18, 2017

Abstract

This work describes the design and test of a localization system called "FeatLoc", usable for robotic applications. It is made up of two subsystems, one using visual odometry and the other visual memory, that is, it searches a global map for known features.

The localization system is designed to operate using features of natural origin, thus providing an alternative solution when artificial beacons are not usable or don't exist. The system requires abundance and variety of features, for this reason, a floor made of natural wood was chosen as the scenery.

The test setup is made up of camera mounted Robot, moving on an indoor scenario of about 2m x 1.5m made of tiles from floating wood. The robot is an adapted RC car platform controlled by an Arduino and commanded by a PC. The PC has an i3 processor, runs Linux 16.04 and is connected to a 640x480 pixel camera. The Arduino runs a ROS node to control the motors. All the code for this system was written using OpenCV and ROS. The camera was mounted so that the image pixel size is 0.8mm or above. The global map is created using images after the camera calibration and homography, resulting in a map resolution of 0.8mm.

Localization is achieved by matching features from the global map to features from the real time image retrieved by the camera. This matching is made using both map and input image after the homography. The feature correspondence process is made up of 3 phases: detection, extraction and matching. The system is able to detect features of the following types: ridges, blobs, edges and corners. Tested detectors are SURF, SIFT, ORB, BRISK and MSER. Tested extractors are SURF, SIFT, BRISK, ORB and FREAK. Used matchers are Brute Force + KNN, FLANN + KNN. Posterior validation of the matches is achieved by ratio test and RANSAC.

In parallel with the previous approach, a visual odometry subsystem was developed to complement the output with relative coordinate information. This subsystem uses the same feature matching architecture (but matching consecutive frames of the camera). Lastly, the system's output is validated using a filter that discards coordinates where the variation is above a certain threshold.

The tests revealed that the SURF/SURF/FB+KNN combined with ratio test and coordinate filter is both the fastest and most robust setting. It allows the global system to run at 10Hz with an accuracy below 3 cm and 4^o orientation.

Resumo

Este trabalho descreve o *design* e teste de um sistema de localização designado "FeatLoc", desenvolvido para aplicações robóticas. É composto por duas abordagens de localização, odometria visual e memória visual.

O sistema de localização é projetado para operar usando características de origem natural, proporcionando assim uma solução alternativa quando as balizas artificiais não são utilizáveis ou não existentes. O sistema exige abundância e variedade de características, por isso, um pavimento feito de madeira natural foi escolhido como cenário.

O ambiente de teste é composto por uma câmara montada no Robô, movendo-se em um cenário interior de cerca de 2m x 1.5m feito de tábuas de madeira. O robô é um carro RC adaptado, comandado por um Arduino e controlado por um PC. O PC possui um processador i3, executa Linux 16.04 e está conectado a uma câmera de resolução 640x480 píxel. O Arduino usa um nó ROS para controlar os motores. Todo o código para este sistema foi escrito usando OpenCV e ROS. A câmera foi montada para que o tamanho do pixel da imagem fosse de 0.8 mm/pixel. O mapa global é criado usando imagens após a calibração e homografia da câmera, resultando em uma resolução de mapa de 0,8 mm.

A localização é alcançada correspondendo objetos do mapa global com objetos da imagem obtida pela câmara. Essa correspondência é feita usando o mapa e a imagem de entrada após a homografia. O processo de correspondência está composto de 3 fases: deteção, extração e correspondência. O sistema é capaz de detetar recursos dos seguintes tipos, cumes, bolhas, bordas e cantos. Os detetores testados são SURF, SIFT, ORB, BRISK e MSER. Os extratores testados são SURF, SIFT, BRISK, ORB e FREAK. Os *matchers* usados são Brute Force + KNN, FLANN + KNN. A validação posterior das correspondências é conseguida por teste de rácio e RANSAC.

Paralelamente à abordagem anterior, foi desenvolvido um subsistema de odometria visual para complementar o resultado com informações de coordenadas relativas. Este subsistema usa a mesma arquitetura de correspondência de objetos (mas usando imagens consecutivas da câmara). Por fim, a saída do subsistema é validada usando um filtro que descarta as coordenadas onde a variação está acima de um determinado limite.

Os testes revelaram que o SURF / SURF / FB + KNN combinado com teste de razão e filtro de coordenadas é a configuração mais rápida e robusta simultaneamente. Permite que o sistema global funcione em com uma frequência de 10Hz e exatidão abaixo de 3cm e 4º.

Acknowledgments

I would like to thank my thesis advisor, Prof. Armando Sousa, as well as the co-advisor, MSc Valter Costa, whose guidance and steering are greatly appreciated. The door to both advisors was always open, whether I ran into a complication or had a question about which methodology to follow or even about writing tips. Both consistently allowed this paper to be steered in the right direction and to avoid future complications inherent to investigations in this area of study.

I would also like to thank my family and close friends for the support. One can easily undervalue this type of support. However the reality is that these people allowed me to focus and regain perspective.

Nolasco Napoleão

“There are four purposes of improvement:
easier, better, faster and cheaper.
These four goals appear in the order of priority.”

Dr. Shigeo Shingo

Contents

List of Figures

List of Tables

Abbreviations and Symbols

BF	Brute Force Matcher
BRIEF	Binary Robust Independent Elementary Features
BRISK	Binary Robust Invariant Scalable Keypoints
DoG	Difference of Gaussian
ESC	Electronic Speed Controller
FB	FLANN Based Matcher
FLANN	Fast Library for Approximate Nearest Neighbors
FREAK	Fast Retina Keypoint
IPM	Inverse Perspective Mapping
KNN	k-nearest neighbors
LIDAR	Light Detection And Ranging
MSER	Maximally stable extremal region
ORB	Oriented fast and Rotated BRIEF
PWM	Pulse Width Modulation
R	Ratio Test
RANSAC	Random Sample Consensus
RaR	Ratio and RANSAC Test
RRe	Ratio Test and Coordinate Rejection Filter
RaRRe	Ratio, RANSAC Test and Coordinate Rejection Filter
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SURF	Speeded Up Robust Features

Chapter 1

Introduction

This chapter raises a few questions to be answered throughout the dissertation and provides a preview of the objectives and scope of the document.

1.1 Scope

In order to achieve higher levels of efficiency in the industrial environment, the decision makers realize that automation is a key component. To automate means to modify a process in order to require less human intervention and less time to deliver. The process of automation implies both increase of speed and decreased human error, which equates to further decline in cost.

Normally the automation is put into place through the installation of automatic equipment, in other words, equipment that is programmed to perform a specific task faster, cheaper and more reliably than the human counterpart. Examples include conveyor belts for easier transportation, intelligent robotic grids for more scalable storage management, vision based sensors for faster quality control, etc.

As an example, in 2012 the multinational Amazon acquired the Kiva Systems for $775 millions. The Kiva systems are no less than a robot army responsible for receiving, depalletizing and storing items, it also transports shelves with items to a packing section. In other words, Amazon bought a storage management system capable of operating with little to no human intervention, in an effort to allow for easier scalability and reduced cost.

This staggering investment shows there is a market for research in the field of indoor localization systems. As such, any method that can provide a faster, easier, cheaper or better approach should be researched.

The Kiva system follows lines as a way to reach the desired destination, thus it can be inferred that the price includes first the manufacturing cost of each individual robot, and second the cost of installing guidelines on the storage facility. Hence, a method not requiring guidelines would get the upper hand, because of the decrease in setup cost.

This research proposes a new application for an old localization paradigm. It provides reduced time and cost for the setup, in addition to reduced cost in infrastructure. The reason is that cameras tend to be less expensive than the industry standard laser-based sensors, such as LIDAR.

1.2 Context

This dissertation was proposed with a simple application in mind: To give a mobile vehicle the ability to localize itself indoors on a handball field. This should be accomplished using cameras as sensing modules.

(a) Handball Field
https://upload.wikimedia.org/wikipedia/commons/d/d5/Pab_central_g.jpg

(b) Natural Wood Sample
https://pixabay.com/en/board-brown-carpentry-dried-grain-1846972/

(c) Granite Sample
https://pixabay.com/en/steinplatte-stone-brown-1331963/

Figure 1.1: Handball Field; Samples for Wood and Granite Texture

In other words, using only visual input information, can the robotic unit navigate and explore the 40x20m field with enough precision to be able for instance to clean it or look for objects, like a ball, inside its boundaries?

The use of vision based localization is very sensitive to the illumination conditions, to the material texture and reflectivity. However due to the fact that the localization is indoor, the illumination only varies slightly, thus removing the necessity of managing the light sources.

1.3 Research Problem

The following questions attempt to summarize the issue that the dissertation proposes to solve for this particular context.

Is it possible to use texture information for robotic localization?

Given enough resolution, can a camera capture differences to uniquely identify a position and orientation?

How robust is this methodology to variations in light or possible occlusion of features?

How fast and precise is this method and how does it compare to the other solutions?

This dissertation studies the use of natural features as a tool for robotic localization. The end product should be a prototype localization system, that is consistent and accurate. Additionally a model for the error distribution of the coordinates would be useful for future comparative analysis.

1.4 Structure of the Document

This document is divided in five chapters, being these the **Introduction** where the scope, objectives, motivations and context are introduced in non-technical terms. Second is **Literature Review** which lists and discusses previous research on this topic and also contains hypotheses to be tested during the dissertation. Third is **Methodology** which contains the core of the thesis, describing the complete architecture of the localization system, as well as individual modules. Moreover it lists and justifies the choices for the test environment and goes in depth in which tests are going to be used to validate the system. Forth chapter is **Results** in which the implications and limitations of the tests are presented. Also it ranks the different settings of the system according to various parameters. The fifth and last chapter is **Conclusion** which offers an opinion-based perspective on the consequences of this research, it also lists future work and possible modifications to the architecture.

Chapter 2

Literature Review

This chapter provides a compilation of the solutions and similar work in this area of study. Additionally the chapter provides a framework for later comparison between the proposed solution and the state of the art.

The proposed solution falls under the category of vision-based localization, as such, the solutions listed on the subsequent section are the ones that fall on the same category. The navigation/ localization problem is not a simple one. For this reason, a general solution is yet to be found. For instance, a method that works indoors may not work outdoors, due to high variation of light conditions. A methodology that is adequate for a factory environment may not be appropriate for office applications due to the presence of people in the floor that end up covering beacons.

Some applications require the localization to be able to find the current position despite being abducted, that is, being moved to an arbitrary location abruptly. Finally the price range may be a limiting factor in adopting a solution. For instance, a rotary encoder is very cheap but not very reliable, on the other hand, a LIDAR localization system is far more robust, resulting in a higher price.

All this considerations will be weighted in to produce a more fair comparison between methods. The vision based methods are divided in three branches. In order to compile the information for this document, the research refers to articles from two initial sources,[1, 2], these serving as surveys and compilations of approaches and results, authored by experts in the field of vision-based navigation.

2.1 Visual Odometry

Visual odometry, also known as optical flow produces coordinates relative to a previous location. Put simply, the displacement of objects within the image frame is proportional to the distance traveled in the real world. This approach finds applications in video compression and motion estimation. Some applications in the realm of motion estimation are altitude tracking for aerial vehicles [3], object trajectory prediction [4], weld seam tracking [5].

This methodology requires a relatively high frame rate, or at least a frame rate high enough to produce consecutive images that are similar, as that is a requirement for accurate results.

As is common in the odometry methods, the accumulation of error throughout the consecutive frames renders the coordinates useless after a few iterations, because of this limitation, an external source of coordinates has to be used to reset the error. Another limitation is the required similarity between consecutive images, which means it is not robust to sudden light variation or abduction.

Visual servoing or motion control via visual input can be used to control a robotic manipulator in order to pick up a moving object [4]. Another article describes the method for calculating the altitude as "tracking feature points through the image sequence" [3]. This article's methodology proposes a good starting point for the architecture of the dissertation: it uses natural features and produces altitude coordinates. The mathematical treatment is very simple, because of the fact that only altitude is calculated, but it provides a strong starting point.

The simplicity of the methodology is very attractive, for this reason, a module using this visual odometry will be added to the architecture of the system, in addition to a second module responsible for fusing the absolute and relative coordinates.

2.2 Robot Navigation using Visual Memory

A second category of solutions is **Robot Navigation using Visual Memory**. Unlike the previous branch, this branch can produce an absolute position: the method simply compares the input image to a representation of the real world in order to produce a correspondence. Once the correspondence is found, the position corresponds to the coordinates within the map. This category can be divided in two approaches **Model based approach** and **Appearance based approach**, the former uses the camera input to recreate a 3D model of the world, the latter uses a "rule" database.

This category of methods finds applications in mobile robot localization and mapping, or SLAM, that simultaneously builds the map and navigates the environment [6]; another application is [7] that uses feature detection and matching to navigate office environment; [8, 9] describe its application in a corridor, with the slight difference that the map is not a complete representation, but instead uses rules to recognize shapes found in said corridor.

Because the method does not rely on the correlation of consecutive image, the frame rate can be lower than the previous method. It is abduction resistant due to the fact that it produces an absolute position, except for the cases of [8, 9]. An important consideration is that the localization is only as good as the real world representation, and for that reason it is important to spend the time to produce an accurate map. Another limitation is that the accidental cover up of a feature can produce undesirable effects.

Model based methods, normally have trouble with perspective due to the irregularity in the object shape. In order to diminish this problem some authors create a 3d model of the map using the input images [6]. That is not going to be the case, as the features are projected on a 2d plane, for the case in study.

The appearance based methods, can navigate without complete knowledge of the environment using rules and sensed information, an example being described in [8, 9]. In the case of corridor navigation, instead of exhaustively listing all the positions, a simple classifier that distinguishes corridor from door and centered position from misaligned is enough to navigate and enter through doors.

Although this approach is better suited for real-time applications, because of the lower processing load, it is not abduction resistant and does not factor the possibility that the environment does not have any distinctive pattern. The robustness and low processing load are desirable traits for a solution.

The most attractive ideology is that of [7], it describes the use of feature matchers to produce localization. The 2d captured image is fed directly to the matcher and the algorithm outputs the closest image. The methodology results in approximately 80% recognition rate, given invariance in perspective and illumination conditions.

This method is designed for an office environment and as such uses the corners of tables, chair, etc. The difference to the proposed method is that the test environment most closely resembles a plane or 2d structure, as such, the method can only rely on texture information.

2.3 Teach and Replay

A third branch is the **Teach replay approach**. It consists of a teaching phase where the program records a successful trajectory through the use of a camera, and then a replay phase where the trajectory is emulated by commanding the robot to emulate the video. Simply described, it consists of having a camera mounted robot being pushed across the desired path and photographing the sequence of images. On the replay stage, the program calculates the difference between the input image and the image from the successful trajectory and uses it to control the direction. The method finds application in robot navigation for indoor/ outdoor environments [10, 11, 12], making it the most versatile branch in robotic navigation.

This method requires no internal representation of the environment, or preprocessing of the image, in the words of this source: "The algorithm is entirely qualitative in nature, requiring no map of the environment, no image Jacobian, no homography, no fundamental matrix, and no assumption about a flat ground plane." [12]. The price to pay for such a versatile method is its fragility to illumination conditions and abduction.

The article [10] shows that the combination of feature based localization and path following produce better results than the pure forms of each. Article [12], uses the same approach but with a neural network-like classification system, making it able to resist the occlusion of some objects as long as enough features are still detected.

This dissertation will not use this methodology, because it lacks scalability, since every new path must be planned.

2.4 Object recognition

Most of the aforementioned methods, describe a preprocessing of the image at some point, either a segmentation, where a corridor is distinguished from a wall, to a feature matching, where visual elements of an image frame are detected, quantified and matched later on to elements on another image. The process of matching frames from different images has been thoroughly researched and the most common solutions are listed below.

2.4.1 Contour Matching

A contour is defined as "the outline of a figure or body; the edge or line that defines or bounds a shape or object" [13]. A contour matcher relies on the invariability of said contour in order to produce a correspondence between images.

Lets take for instance a square rotating around its center on a 2d plane (Figure 2.1). For an observer situated on top of the plane, the object retains its shape, number of vertexes and dimensions despite the rotation.

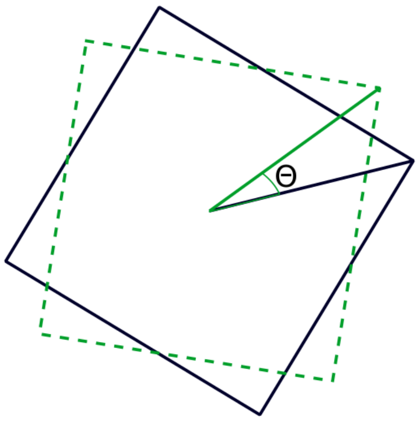


Figure 2.1: Illustration of Rotating Square

This method can be described in 3 steps, detection of shapes on the image frame, computation of a description for each shape and matching of the shapes.

The detection is normally done using a canny edge detector [14], that converts the colored image to a black and white one with only the edges marked. This step is followed by a linking in which the edges are closed in order to obtain shapes that delimit areas and not open curves.

The computation step, in which a shape is assigned a number according to its characteristics, can be done using the invariant moments or fourier descriptors. The invariant moments are mathematical expressions that input the contour and produce a decimal value that describes the shape. The Hu moments [15] are a set seven moments invariant to scale, translation and rotation. The Fourier descriptors [16] tend to be more robust than the invariant moments and therefore finds applications in gesture recognition software. This computation should be as much as possible invariant to rotation, scale, translation, shade. Ideally the only parameter that matters is the shape of the contour.

The final step of matching is done by brute force, it consists of comparing the Hu moments error between contours and returning the contour that minimizes the error. This error can be calculated using several different metrics [17]. Even though the Fourier descriptor is more robust, no attempt will be made to incorporate it in the system architecture, as the OpenCV library does not have a stable documented version of this function, and the schedule does not allow time to be spent in its implementation.

Considering the invariance to rotation, and translation, in addition to its suitability for 2d environments, contour matching will be explored on the dissertation.

2.4.2 Template Matching

Template is defined "as something that is used as a pattern for producing other similar things" [18]. This approach uses a brute force comparison between input image and an internal representation of the world.

The template matcher is a function supported by the OpenCV library, the algorithm works as follows: the input image is slided along the x and y axis; for every x and y the error is computed as a sum of the distance between each individual pixel. In the end the smallest sum is considered to be the matching position.

This approach is very susceptible to error, because it focuses on pixels instead of patterns, in addition it is very computationally heavy as it is a brute force pixel by pixel search, and that does not even factor different rotations or scaling. On the other hand template matching is more suited for an environment where the illumination conditions don't differ between the input image and internal representation.

All the information on the feature matcher algorithm implemented in the OpenCV library was consulted on [19].

2.4.3 Feature Marching

A feature is any interest point that has visual predominance in an image. In the computer vision realm a feature can be any of 4 shapes, blobs, edges, ridges or corners. This approach can be described in 3 steps, those are: feature detection, feature description and feature matching.

2.4.3.1 Detector

The detector lists features wherever the difference of gaussians (DoG) exceeds a certain threshold, every entry contains a set of coordinates, the angle and the size of the feature. The list can be obtained using various detectors.

The detectors supported by OpenCV are SURF, SIFT, ORB, BRISK and MSER. The main difference among them is the threshold of detection, or the type of feature that is searched (blobs, edges, ridges or corners).

A typical architecture for detector includes the application of a Harris corner detection [20] or blob detection algorithm to obtain the initial set of features. This step is followed by a successive DoG (difference of gaussians) [21] and subsequent sum of the differences that enhance the "best" features. The feature detectors vary in the threshold of detection for the final step or in the DoG. The remaining of the section presents a historical overview and compares the detectors.

BRISK stands for Binary Robust Invariant Scalable Keypoints and was proposed by Leutenegger et al. (2011)[22] in an attempt to reduce the computational cost of feature detection, being ideal for real time applications running on low power devices.

ORB stands for Oriented fast and Rotated BRIEF and it constitutes a fusion of the FAST [23] and BRIEF [24] detectors. The initial detection of features uses the FAST detector, being then followed by a filtering based on the Harris corner measure. This detector is rotation invariant and resistant to noise, born out a necessity for faster detection, for use in real-time applications was first introduced by Rublee et al. (2011)[25].

SIFT stands for Scale-Invariant Feature Transform and it is a patented feature detector and extractor first introduced by D. G. Lowe (1999)[26]. This detector offers improved accuracy, stability and speed as well as scale and rotational invariance.

SURF, Speeded Up Robust Features, is also a patented feature detector and extractor inspired by SIFT. It was proposed by H. Bay et al. (2008)[27]. The architecture of the method allows for a configurable threshold in the detection of features. The improvement over the SIFT is a speed increase and alleged increased robustness in addition to the scale and rotation invariance.

MSER stands for Maximally stable extremal region, developed by Matas et al. (2002)[28], as an attempt to create a feature detector invariant to viewpoint. It presents great robustness to light variation and is both scale and rotation invariant.

2.4.3.2 Extractor

The objective of this component is to describe each detected feature as a non repeating scalar. Each scalar should describe uniquely the feature, but at the same time group together features with the same color, shape, shade, etc. despite rotation, scale and noise.

The extractors supported by OpenCV are SURF, SIFT, ORB, BRISK and FREAK. The extractors differ in the number of bits that are reserved to describe a pixel and the level of invariability for each parameter (shade, contrast, noise, etc).

FREAK stands for Fast Retina Keypoint, it was inspired by the human visual system. First proposed by Alahi et al. (2012)[29], it attempted to maintain robustness while reducing computational cost in an attempt to incorporate feature matching in smart phones.

2.4.3.3 Matcher

The matcher is the last step of the architecture, it is responsible for receiving 2 lists of features with the corresponding descriptors and establishing correspondences. The correspondence is made by minimizing the error function between different features, the OpenCV library providing several metrics for the error calculation.

OpenCV provides two search modes: Flann Based Matcher and Brute Force Matcher. A brute force search always returns the ideal match. Flann stands for Fast Library for Approximate Nearest Neighbors, this method was developed to provide a faster option, for this reason the result is not ideal or as consistent as the brute force. Finally the OpenCV library has the option knnMatch that list the k best matches for each feature.

2.4.3.4 Match Outlier Rejection

Ideally only the real correspondences should be made by the matcher, that is not the case often. What happens is that the features are matched to other features incorrectly, and as such need to be filtered, that's where the knnMatch is useful. After the matching is done, the next step is to discard the outliers, meaning the matches that do not correspond to the real features. A quick literature review showed two approaches to filter the matches, ratio test and RANSAC test.

The ratio test consists of filtering matches where the error is at least r times less than the next best match. In other words, the second best match should not be as good as the best match.

RANSAC stands for RANdom SAmple Consensus, it is an iterative method used to estimate the parameters of a mathematical model. The method works by choosing a representative amount of points out of all the matches and calculating the smallest degree model that is representative of that input. In other words, the test returns a smaller number of matches as inliers, in a way that is still representative of the input.

Chapter 3

"FeatLoc" System and Test Platform

This chapter will include the overall system architecture as well as all methods and assumptions necessary to have a working test platform. The various steps and nodes of the system will be described and explored. The nodes include an image capture node; a node to remove image distortion; a node to calculate the coordinates. Most programs and functions were written in C++ using the ROS framework, as such it is important to understand its operation and terminology before advancing (see annex A).

3.1 Architecture Schematic

Figure 3.1 shows a schematic of the overall architecture, along with a summary of every node. The full description is available in the corresponding section.

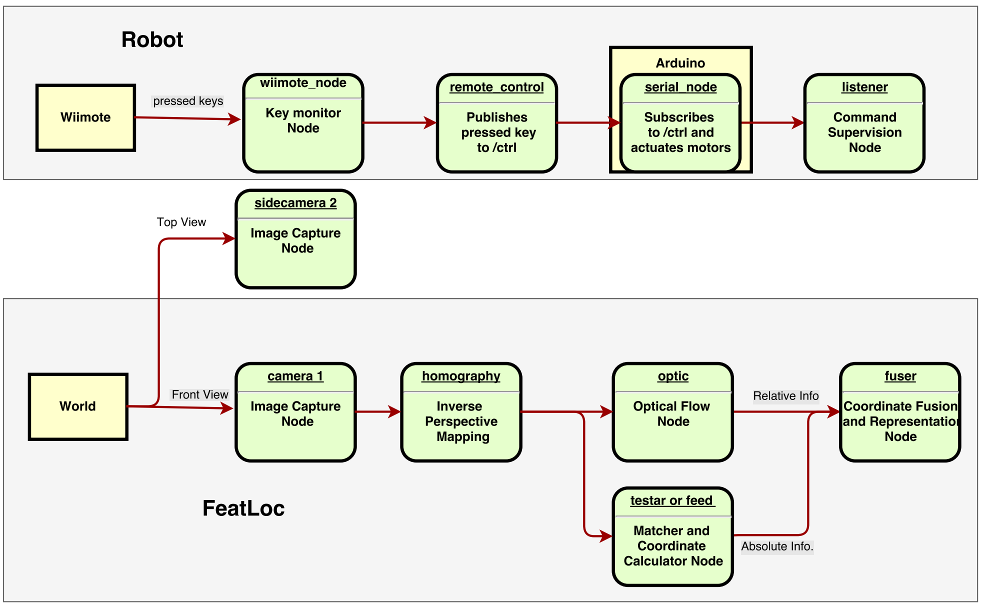


Figure 3.1: Solution Architecture

The resulting system is composed of two independent subsystems. The first is responsible for **navigation**, the nodes in this subsystem monitor the Wiimote state, publish a message when specific keys are pressed and control direction or speed according to input messages. The second subsystem is the **localization subsystem**, responsible for acquiring the input image from a frontal camera; correcting the frames for lens distortion and perspective; matching features and calculating coordinates. The navigation subsystem is not indispensable to the localization, but is a tool to facilitate image acquisition.

The ROS framework has a funtion rqt_graph, used to visualize the connections between nodes. Figure 3.2 was produced by this tool in order to validate the system architecture. Equivalent ot the system architecture, three independent subsystems are represented, with the upper one being the top view camera, the middle one is the FeatLoc and the last one is the navigation system.

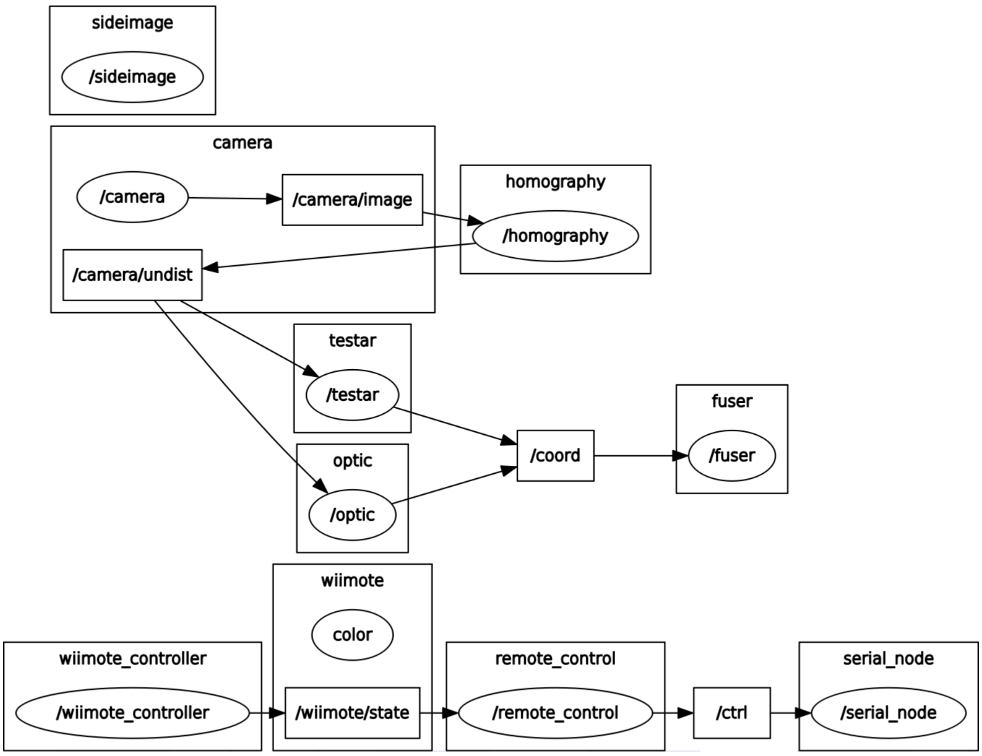


Figure 3.2: rqt_graph of system architecture

3.2 Navigation Subsystem

Having in mind the initial application, the handball field, the dimensions would result in several weeks of mapping and testing if done exclusively through human labor. As such, this dissertation chose to create a remote controlled platform that could be tele-operated to allow remote video capture.

3.2.1 Traxxas Eletric Vehicle

The ***Slash 4x4*** is a radio controlled 4x4 electrical truck developed for all terrain by traxxas, built using Ackerman steering geometry.

Due to its availability in the FEUP laboratory, and wide frame, it became a viable candidate for the mobile vehicle. The wide frame provides a platform for the computer and cameras.

3.2.1.1 Mechanical Adaptation

The radio receiver/ emitter module sold with the car was not found in the vehicle, meaning a reconstruction of the control module was necessary. Also, in later stages of the project, an automatic trajectory follower could be necessary, for that reason it was important to incorporate a computer controlled navigation system.

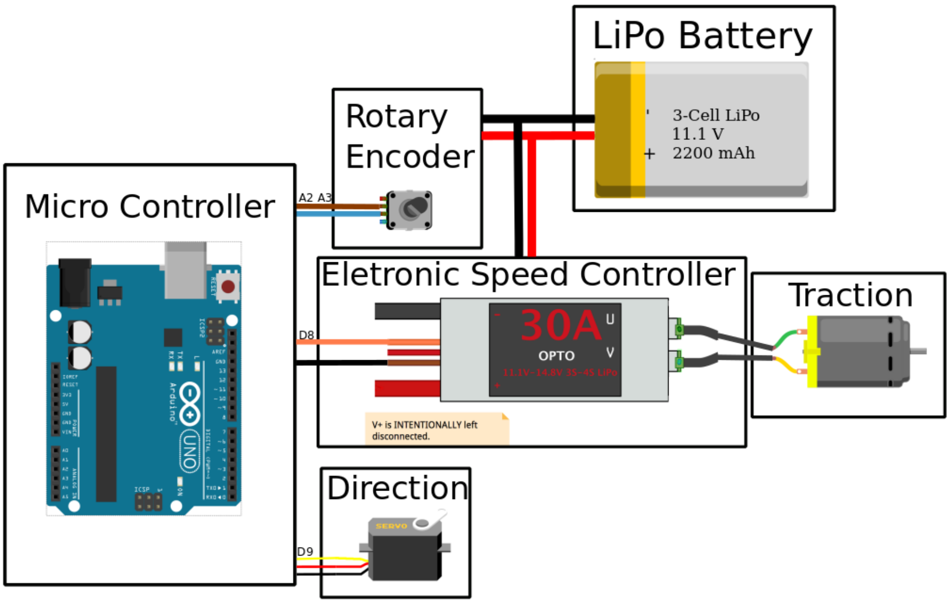


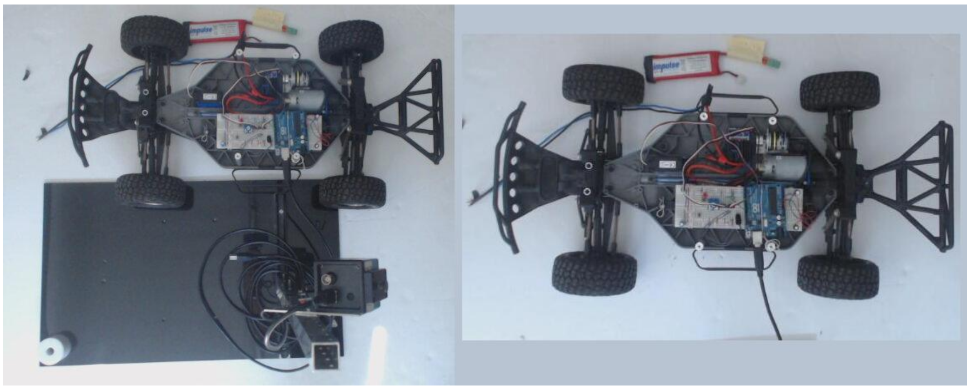
Figure 3.3: Illustration of electrical connections

The first step was then to install a micro controller (Arduino) where the radio module would be found. The Arduino would produce the necessary control signal, in the shape of PWM (Pulse-width modulation), to guide the vehicle through the environment, both direction (port D9) and traction (port D8) being commanded by the arduino.

Both direction and traction motors were supplied by the RF module, so the lack of the radio receptor required a rebuild of the power-supply channels. Steering became supplied by the Arduino's power supply, and the traction would be supplied by an external power supply, in this case, a two cell LiPo 11.1V (3*3.7V).

Finally, a rotary encoder was added to the vehicle. The rotary encoder has 2 outputs/ phases (connected to port A2 and A3) that produce rising/ falling edges according to the wheel's position.

The connections are represented in illustration 3.3, whereas the final result in the car is in 3.4a.



(a) Eletrical circuit

(b) 3/4 view and Side view

Figure 3.4: Adapted RC Car

The camera position had two requisites: it had to be close enough to capture the small nuances in texture, but distant enough to allow for the widest area to be captured. The end result is observed in 3.4b.

3.2.2 Serial Node

Once the mechanical adaptation of the car was completed, an interface with the computer had to be implemented. This node runs in the laptop on top of the car. This is responsible for subscribing to the messages with topic "/ctrl" and controlling the motors according to the messages. The messages with this topic are published by the wiimote_node and have 5 types:

- u (up pressed) - moves forward.
- d (down pressed) - moves backward.

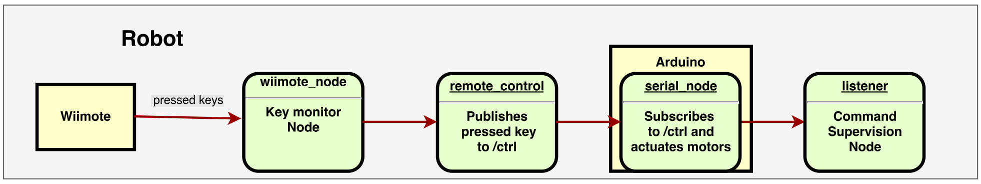


Figure 3.5: Navigation Subsystem

- r (right pressed) - rotates clock wise.
- l (left pressed) - rotates counter-clock wise.
- s (button 1 pressed) - sends a calibration signal.

In order to operate the RC truck, the manual specifies that the Electronic Speed Controller (ESC) must be turned on, followed by the remote in neutral position for 3 seconds, once the ESC shows a green light, it means the motor can start moving. Due to the fact that the Radio Control module was removed, the neutral signal was simulated using the Arduino. The s message triggers the output of the calibration PWM.

The calibration signal was chosen at random, and set at 1000uS, the manual did not specify any value and that value proved appropriate. After setting the neutral level, the upper and lower limits that produce movement were searched through trial and error.

Table 3.1 shows the duration of the PWM, following the trial and error. Load and without load have different control signals because the increase in weight translates to a longer PWM's period. The table reads as follows. In order to produce forward motion without a computer on top, the servo motor for direction receives a 94ms PWM and a 1660us (1475+185) PWM for the traction motor. If the traction motor receives a PWM of 1475, the motor will not move. Any value higher than 94+41=135ms exceeds the torque capability of the direction servo, causing it to force the motor. The step of the motor is the variation chosen for the duration of the command signal, this is necessary to guarantee a gradual increase or decrease of the PWM period; if the motor goes instantly from 1475 to 1600, the ESC is programmed to ignore the PWM, resulting in failure to move.

The second step of the navigation subsystem is performed by a function running in the Arduino. Every time a message is received, a PWM is sent to the motors, the duration of the PWM being limited to avoid anomalous behavior. At the start of the PWM, a decreasing counter with 8 cycles is initiated, once the counter reaches zero, the control signal returns to the neutral level. This approach was preferred to a differential speed control because in case the node disconnected or malfunctioned, the car would stop in 8 cycles (less than a second).

The initial plan to have odometry on the wheels failed to be implemented, due to the interrupts of the Arduino. In order to count the transitions of the rotary's phases, an interrupt would be programmed for each phase, however because the interrupt stops the output of the PWM, the navigation and odometry are incompatible.

Table 3.1: Trial and error constants for traction motor actuation

(in microseconds)	Neutral Level	Smallest deviation that produces traction		Step	Calibration
		Foward	Backward		
Frame without laptop	1475	185	-85	25	1000
Frame with laptop		450	-140		

Table 3.2: Trial and error constants for direction servo actuation

(in miliseconds)	Neutral Level	Max allowed deviation for rotation		Step
		Clock Wise	Counter Clock Wise	
Direction	94	41	-41	1

The implementation of rotary encoder monitoring for Arduino is based on the code from this site[30].

3.3 Localization Subsystem

The localization subsystem requires three things, a node to remove lens distortion and perspective, a working map of the world and a node to calculate the coordinates.

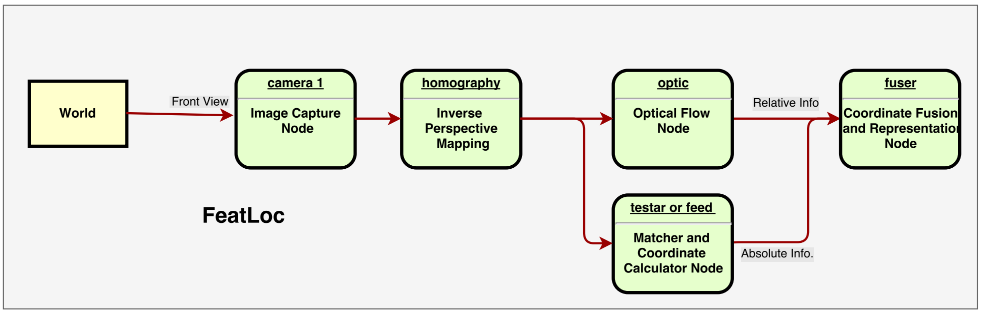


Figure 3.6: FeatLoc Subsystem

3.3.1 Homography Node

This node's purpose is to receive the captured frames from the camera node and produce an image that appears to be captured from an aerial view. The node has two functions, the first is to correct the distortion added by the fish eye lens and the second is to remove the distortion added by the perspective. The steps for the algorithm can be described as follows:

First step is to correct the barrel distortion, that causes straight lines to appear curved. The correction is applied by multiplying the input image matrix by a camera matrix and a distortion coefficient vector. The camera matrix and distortion coefficient vector were calculated using this toolbox [31]. The toolbox inputs images with the chessboard pattern from different perspectives, and the manual input of some vertexes positions. It then outputs the matrix that corrects the barrel distortion.

Second step is to correct dimension and perspective distortion, which causes a square to appear as a trapezium. Ideally the input image should appear to be coming from an aerial camera facing down. In order to produce this effect, the input image is stretched horizontally on the upper edge, to account for the perspective.

A demonstration of the algorithms result is found in 3.7.

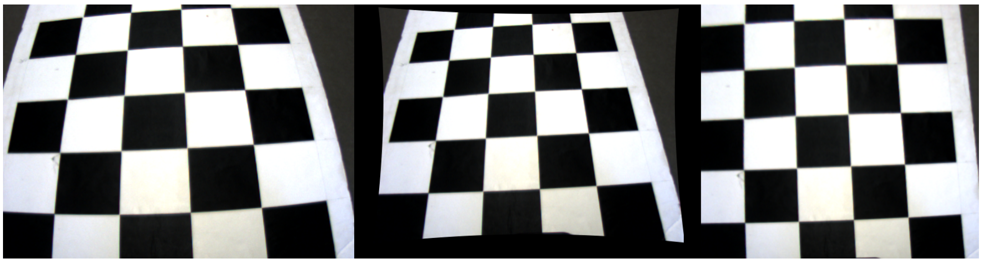


Figure 3.7: Left to right: input image, barrel distortion removed, after homography

3.3.2 Map Construction

The idea of image matching only works if a representation of the environment is available. For the purposes of output validation, a map that represents exhaustively all the space is used. Alternatively a list of features and respective coordinates would suffice. Ideally the input image and map should have the same resolution, and the map must also represent as accurately as possible the real world proportions of the objects.

The test environment is a 217x138cm wooden rectangle. The rectangle was built using boards on top of each other. Several considerations were taken in its construction. First the boards were not to repeat themselves in order to remove the ambiguity of having several matches, as such any repeating board was discarded. Second the test environment should resemble a handball field, or at the least should have a simple shape (square, rectangle, adjacent rectangles). Third, the wooden planks should remain clean, or at least uncluttered during the test phase.

In order to map the test environment, as much as 50 images were saved. All the images were saved after the inverse perspective mapping node and cover different parts of the environment, with enough overlap to allow stitching, but at the same covering the greatest area possible.

The stickers served various functions during the construction of the map. First, they helped to delimit the area that each image should cover, second they automated the subsequent stitching of the images. In a latter phase of the process the stickers were removed as they would produce matches with artificial features, thus defeating the objective of the dissertation.

Initially the plan was to use the stitcher functions from OpenCV. However since the images resemble aerial views of the floor, the stitcher would have to use mosaic stitching that is not supported by OpenCV. The OpenCV library only supports spherical stitching, which produces a map that appears stretched over a sphere, with the corners out of place.

The map was created using Hugin - Panorama photo stitcher[32], a free cross-platform stitcher. The process of stitching the map follows several steps. First the images are added to the program, second the images are moved, close to their final position, by the user, then the control points for the stitching are added using an automatic detector. Finally the program produces the map, and the user validates the map visually, in case it is defective, new control points are added manually.

The map is valid if the wooden planks are continuous and rectangular, the lines in the contour are parallel and the shade is uniform.

The resulting map is shown in 3.8a. As image 3.8b shows, the areas in white signal areas with the highest probability of error. In order to calculate those areas, the idea was to look for broken edges that occur when there is incorrect stitching and positions far from boundaries. The boundaries are easily recognizable and have a continuity that guarantees a correct alignment of the coordinates, for this reason, the further a position is to a boundary, the highest is the probability of error. Figure 4.2 shows the test environment as it is seen by FeatLoc, the stickers are not used for localization.

Three attempts were made to use visual memory as a localization option, contour matching, template matching and feature matching. Only the last was considered a viable option.

3.3.3 Contour Matching

Contour matching's structure is as follows, first the map and input image have a threshold applied, then the lines are thinned and linked using the Canny edge detector, afterwards the outer contours are described using the seven Hu moments, finally the contours are matched between images.

All shapes outside the image frame are discarded, otherwise they would appear as another complete shape, also any nested shapes are removed, as those are generally noise from the image capture and too small to be used as matches.

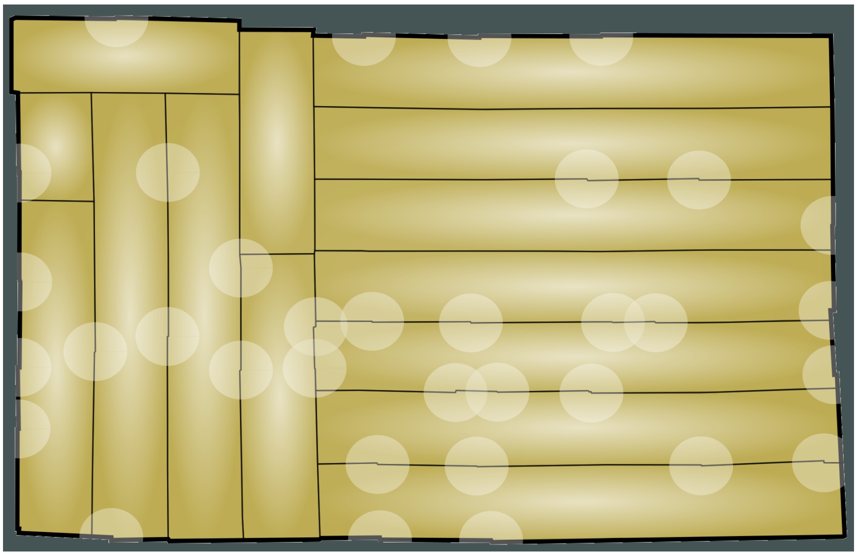
The results for two different materials are shown in 3.9a and 3.9b: the first image shows a great amount of detail, but proves very sensitive to noise and shade, the second does not provide detail on the tile interior.

The idea of using this method was abandoned due to the high sensitivity to illumination conditions. Since the contour is only detected if limited or closed, the incidence of light became a problem, as it caused the contours to become deformed.

3.3.4 Template Matching

Similar to the previous approach, this method was discarded for two reasons. One was the computational cost of rotating and matching the rotated image which makes the solution infeasible. The other was light susceptibility: the template matcher uses color information, so for this reason the

(a) Stitched map



(b) Error zones signaled with white

Figure 3.8: Representations of the test environment

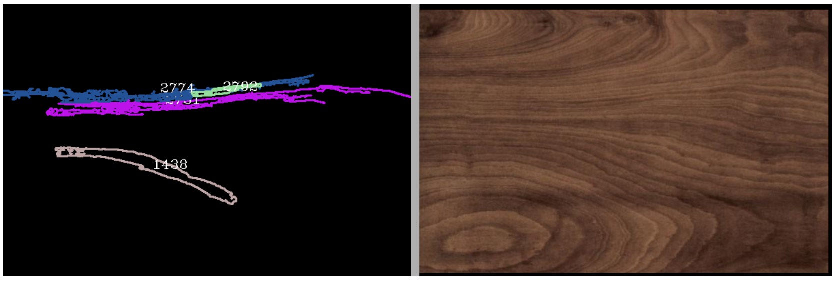
light conditions have to be the same in both images, meaning that a casted shadow would make the coordinate calculation diverge.

The result for this function is shown in 3.10: as shown the program return an incorrect correspondence, one possible cause being the shadow present in the inferior left corner.

The code developed for this node was based on the code from this link [19].

3.3.5 Feature Matching

The feature matcher node uses the code available here[33] as a reference.



(a) Contour detection for wooden boards

(b) Contour detection for tiled floor

Figure 3.9: Results for Contour Matcher

3.3.5.1 Feature Detector

This step receives as arguments the undistorted input image and the map. It outputs a list of key points or features, each having the coordinates and the angle of its center of mass as well as the size of the feature. The program supports SURF, SIFT, ORB, BRISK and MSER as detectors, and can change the detector by changing the detector iterator in the user interface.

3.3.5.2 Feature Extractor

This step has as inputs the key point lists from the previous step. The extractor produces a list of scalars with the same length as the key point lists. The program supports SURF, SIFT, ORB, BRISK and FREAK as extractors and can change between extractors by altering the extractor iterator in the user interface.

3.3.5.3 Feature Matcher

This step has as inputs the lists from the previous steps and outputs a list of matches. The options for the matcher are brute-force and flann-based, which can be accessed by altering the matcher iterator in the user interface.

K Nearest Neighbours (Knn) is a setting that outputs k matches for each keypoint. K was set to 2, for the purposes of filtering outlying matches.

Posterior to the matching, the matches have to be validated, this being done by discarding the outliers, meaning the features that do not correspond to the real position on the map. A quick literature review showed two approaches to filter matches, ratio test and RANSAC test.

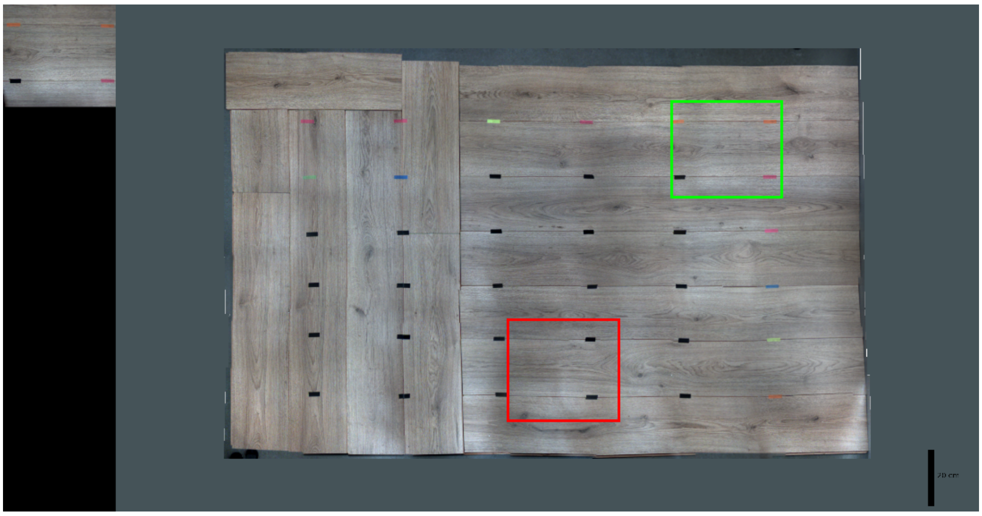


Figure 3.10: Template matcher: Green (correct match) and red (calculated match)

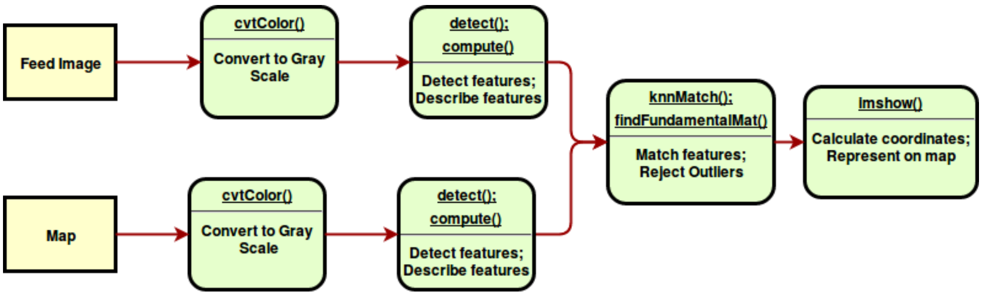


Figure 3.11: Fluxogram for feed node

RANSAC is implemented by a function from the OpenCV library, which receives the matches from the matcher node and calculates which ones are inliers. A match is an inlier if its coordinates are representative of the complete distribution of matches. This filter is not recommended if the outliers outnumber the inliers.

Ratio test only works for the matches obtained with knn matcher for k>=2, meaning each feature has at least 2 matches. The filter discards any match that is not at least r times better than the second best match. The remaining matches are put in a vector called "good_matches" that is used for coordinate calculation. This test is compatible with the RANSAC test.

After many attempts at filtering matches, the test shows that the ratio test may be applied alone, or combined with RANSAC, but the RANSAC cannot be use alone, because it fails to remove enough outliers.

3.3.6 Coordinate Calculation

The match class in OpenCV, has a distance variable that corresponds to the error between features, also it has a train and scene variable with the coordinates of the matches as found in the input image and map respectively. This function determines the coordinates using as input the good_matches vector obtained after the validation step.

Image 3.12 represents the relation between map and feed coordinates. The green dot present in the map and input image represents the front of the car, and ultimately the position of the robot. The variables rows and cols, present in the field of view, are the number of rows and columns of the feed image. The trapezium on the right image represents the field of view before the homography.

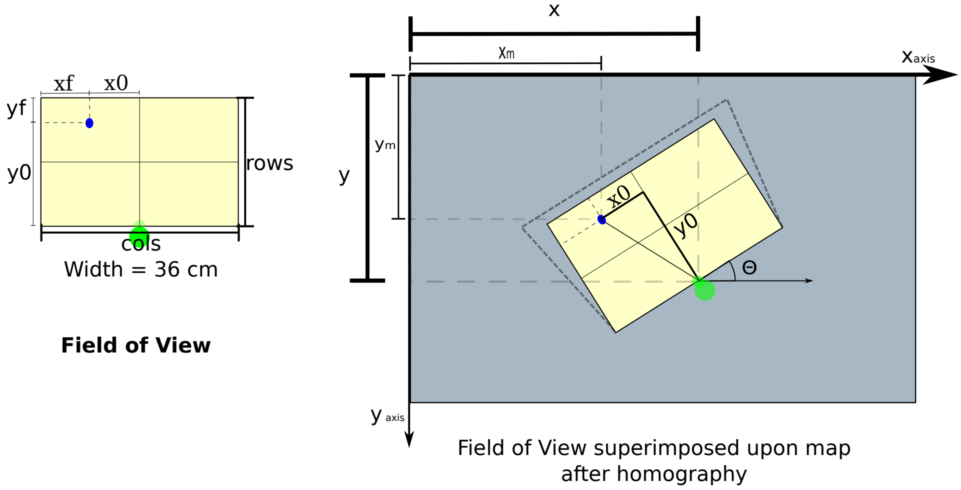


Figure 3.12: Schematic for coordinate calculation

M (equation 3.1) stands for mass, and is given by the inverse of the error between features. The mi variable corresponds to the inverse for the error between the map and feed feature for match i. Variable N represents the length of the match list.

$$M = \sum_{i=1}^{N} \frac{1}{m_i} \tag{3.1}$$

The first step is to calculate the weighted average for the coordinates both in the input image and map. The variables with suffix m or suffix 2 correspond to the map image. For instance variable x2i corresponds to x coordinate of a feature in the map image.

$$x_m = \frac{1}{M} \sum_{i=1}^{N} \frac{x2_i}{m_i} \tag{3.2}$$

$$y_m = \frac{1}{M} \sum_{i=1}^{N} \frac{y2_i}{m_i} \tag{3.3}$$

Suffix f or suffix 1 corresponds to the feed or input image.

$$x_f = \frac{1}{M} \sum_{i=1}^{N} \frac{x1_i}{m_i} \tag{3.4}$$

$$y_f = \frac{1}{M}\sum_{i=1}^{N}\frac{y1_i}{m_i} \tag{3.5}$$

The orientation of the robot is calculated by the weighted average of the differences between the feed image and map.

$$\theta = \frac{1}{M}\sum_{i=1}^{N}\frac{\theta_i - \theta 1_i}{m_i} \tag{3.6}$$

It is important to notice that variable θ has several representations $\theta; \theta+2\pi; \theta-2\pi; ...$ because it is an angle. If we take for instance 360º and 10º, we calculate a difference of 350º, when the real value is 10º. In order to avoid this problem, all the angles are mapped in order to approach the value of an initial seed or anchor. The anchor is the first angle obtained for the vector of good matches, for instance vector $\{350, 12, 40\}$ becomes $\{350, 372, 400\}$. A second precaution is to map the differences between 0 and 360º, which means a difference in orientation of 370 is equivalent to 10.

Coordinates with suffix f are converted to suffix 0 to account for the incorrect orientation of the x and y axis. The cols and rows variables are respectively the width and height of the input image.

$$x_0 = \frac{cols}{2} - x_f \tag{3.7}$$

$$y_0 = rows - y_f \tag{3.8}$$

Finally, the output coordinates of the front of the robot are calculated by subtracting the position of the object by the object's postion on the map.

$$x = x_m + x_0 \cdot cos(\theta) + y_0 \cdot sin(\theta) \tag{3.9}$$

$$y = y_m - x_0 \cdot cos(\theta) + y_0 \cdot sin(\theta) \tag{3.10}$$

3.3.7 Visual Odometry

The optic node's compares the current input image to a previous one. The node uses the same architecture as the feature matcher, that is SURF detector, SURF extractor, knn brute-force matcher and RANSAC/ratio test for outlier rejection. Finally the displacement calculation follows the same architecture as the visual memory with the difference that the map is replaced by a previous input image.

As before, the first step is to calculate weighted coordinate average for the set of matches. The variables with suffix 1 correspond to the previous image, while the suffix 2 corresponds to the current image. M stands for mass, and is given by the inverse of the error between features.

$$M = \sum_{i=1}^{N}\frac{1}{m_i} \tag{3.11}$$

$$dx = \frac{1}{M}\sum_{i=1}^{N}\frac{x2_i - x1_i}{m_i} \tag{3.12}$$

$$dy = \frac{1}{M}\sum_{i=1}^{N}\frac{y2_i - y1_i}{m_i} \tag{3.13}$$

$$d\theta = \frac{1}{M}\sum_{i=1}^{N}\frac{\theta 2_i - \theta 1_i}{m_i} \tag{3.14}$$

Again, it is important to notice that variable θ has several representations $\theta; \theta + 2\pi; \theta - 2\pi; ...$, because it is an angle, so in order to obtain consistent results for the difference, all the values are "anchored" around an initial value. For instance if the program produces $\{350, 12, 40\}$ as results, they become $\{350, 372, 400\}$.

3.3.8 Coordinate Information Fuser

The fuser node's objective is to subscribe to the **coor** topic and display the vehicle position overlapped on the map. Messages from the **coor** topic are published by the optical and feed nodes with have 2 different types. Either a differential coordinate calculated by the visual odometry node or an absolute coordinate by the feed node. The message is composed of the dx, dy, dθ fields corresponding to the increments in each dimension. The current coordinate is equal to the previous coordinate added to the increment.

In this case the new coordinate position replaces the previous coordinates. Figure 3.13 showcases the result of the different localization approaches and combinations.

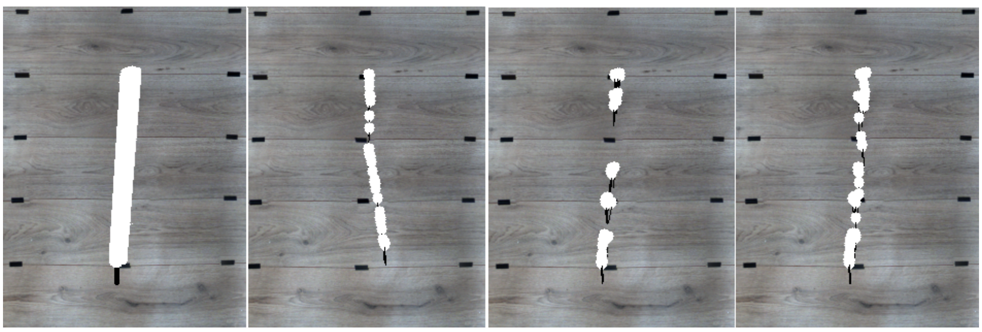


Figure 3.13: Left to right: Real trajectory, Trajectory via odometry, feature matcher and fusion

After the coordinates are calculated, any output that is off-key should be discarded. Any variation in orientation or distance between consecutive coordinates that exceeds a certain threshold is filtered at this time. The value chosen for the distance threshold is 20 cm and 10^{o} for the angle and was chosen through empirical observation.

3.3.9 Test Node

The test node was written as an attempt to reduce the huge overhead of the feature matching algorithm. The final version of the feed node produced results every 1.5 seconds, the *test* node was designed to reduce this time at least in half.

The feed node is not designed for efficiency of computational resources. The objective was to have a responsive, simple interface to swap between different methods and parameters.

The typical feed cycle includes reading the map image (approximately 7Mb), receiving the input image, converting both images to gray scale, proceeding to the detection, extraction and matching of features, removing the outliers and calculating the coordinates.

Since map image does not change, it is not necessary to read, detect and extract features in every cycle. After removing this unnecessary steps the average processing time dropped to 200 millisecond.

This node was assigned the task of gathering statistics, such as average processing time, average error, error standard deviation, etc.

Chapter 4

Results

The tests from this section, had the objectives of finding the best combination detector/ extractor/ matcher, determining the robustness of the localization subsystem to noise, variant illumination and rotation.

The SURF detector has a customizable parameter, called min Hessian, that was set to 200. This value was recommended by users working on similar applications and proved to work under various brightness conditions.

In order to have repeatable results the ROS platform allows the user to replay a sequence of messages in the same conditions that they were acquired. This functionality allows the programmer to have repeatable results, that is, different combinations of parameters can be tested for the same exact input.

4.1 Tests

In order to validate the localization algorithm the tests require a quantification of the error, and a model of said error. Error is described as the difference between the real value and the measured one.

The real coordinates have to be obtained using an external, more reliable source, such as a laser based localization system or even a measuring tape. The initial idea was to use UbiSense as the ground truth, however due to several complications it was replaced by a measuring tape.

4.1.1 Sequence without Movement

Since the measuring tape is unable to produce real coordinates in real time while moving, the tests were done using a sequence of frames, or bags, where the robot is not moving, in a known position. The test was done for 6 different locations, changing the position, angle and illumination in order to quantify the robustness to these conditions. The test locations were chosen in order to be representative of the map. The bags average around 7-10 seconds and have 25Hz frame rate.

Figure 4.1 shows the characteristics of each bag used for the tests.

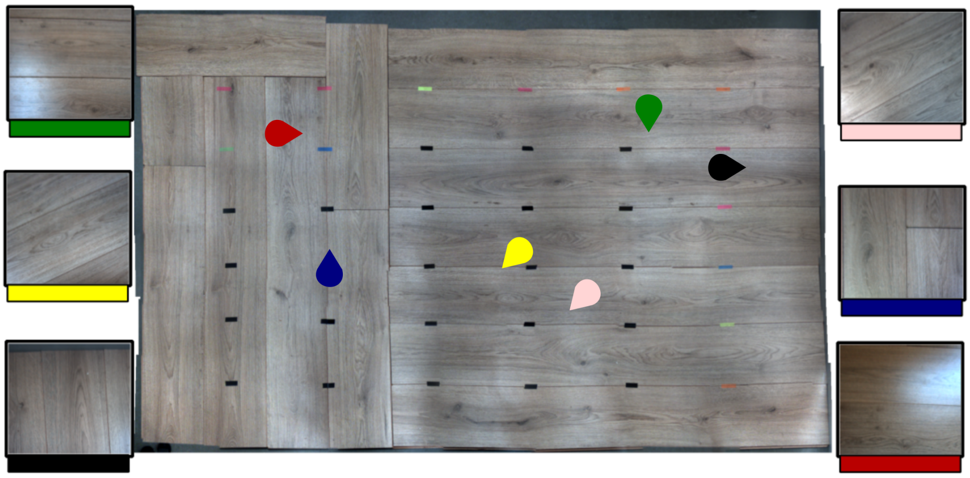


Figure 4.1: Position and orientation of the bags

4.1.2 Trajectory Detection

Even though the error cannot be quantified, this test serves only as a demonstration or proof of concept. The validation is visual and consists of a second recording from an aerial view. Figure 4.2 shows the result for the detection of a descending trajectory.

4.1.3 Tested Parameters

Nine parameters were chosen in order to compare the combinations of detector/ extractor and matcher, they are:

Processed Images - Corresponds to the number of images that the program was able to process. It is proportional to the speed of the processing cycle.

Localization calculated - Corresponds to the number of times the program had more than 1 match and could calculate the position.

Localization not filtered - Corresponds to the number of times the filter of coordinates did not discard the position.

Localization correctly calculated - Corresponds to the number of times the position obtained was within a certain range of the correct response.

(Precision) Standard Deviation x, y and theta - indicates how close the coordinates are to each other.

(Accuracy) Standard Deviation or Average Error for x, y, theta - indicates how close the coordinates are to the real value.

Average matches per image - Corresponds to the average amount of matches found for each image after filtering outliers.

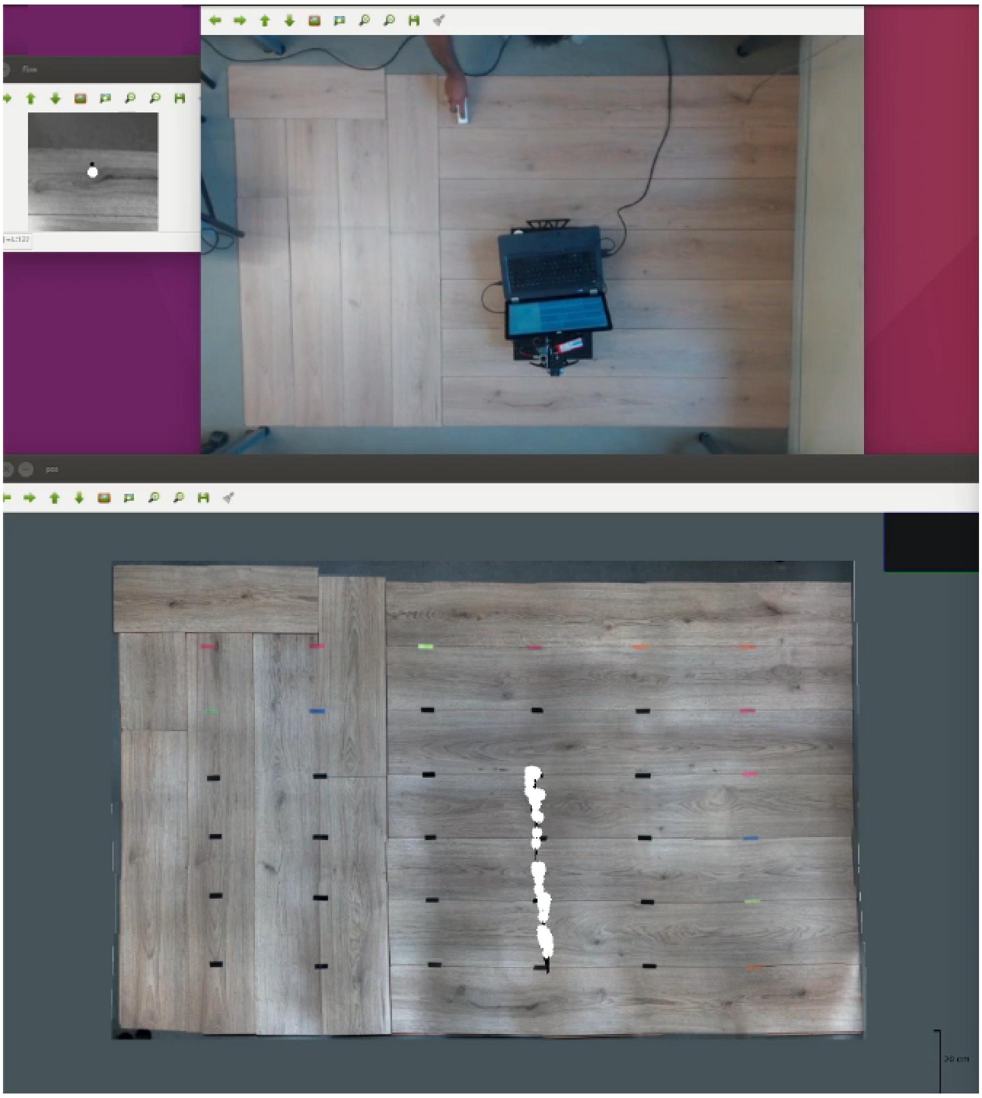


Figure 4.2: Aerial view (top) and calculated trajectory (down)

Average processing time - Corresponds to the average duration of a processing cycle.

4.1.3.1 Average Error

The average error is described as the mean deviation between the real and calculated coordinates.

The real coordinates are obtained by rotating/ sliding the input image and overlapping the result on the map. A second validation using the measuring tape assures the orientation was measured correctly and the coordinates match the real world coordinates.

The error caused by the multiple representations of the angle occurred again, and was dealt as before.

4.1.3.2 Error standard deviation

This assumes that the output coordinates are spread close to the real ones, and that the error follows a normal distribution.

The accuracy measures how close the measured values were to the real one. The precision measures how close the measured values were to the mean of said values.

The tests for accuracy and precision differ only in one term. The formula for accuracy is,

$$\sigma_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N}((x_{real} - x_i) \cdot res_x)^2} \tag{4.1}$$

$$\sigma_y = \sqrt{\frac{1}{N}\sum_{i=1}^{N}((y_{real} - y_i) \cdot res_y)^2} \tag{4.2}$$

$$\sigma_\theta = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\theta_{real} - \theta_i)^2} \tag{4.3}$$

The precision of the model requires the average value to be determined previously, and is calculated by:

$$\sigma_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N}((x_{avg} - x_i) \cdot res_x)^2} \tag{4.4}$$

$$\sigma_y = \sqrt{\frac{1}{N}\sum_{i=1}^{N}((y_{avg} - y_i) \cdot res_y)^2} \tag{4.5}$$

$$\sigma_\theta = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\theta_{avg} - \theta_i)^2} \tag{4.6}$$

The multipliers res_x, res_y are respectively the resolution of the camera along the x and y axis. These scalars converts the standard deviations from pixel to millimeter.

4.2 Test 1 : Number of Features

This test calculates the number of detected features for the map image. The reader should have in mind that the map is a 3000x2000px png image, corresponding to 138x217cm or $3m^2$ in the real world.

The ability of each detector to find features will directly affect the performance of the combination. This test shows SURF as the undisputed best detector averaging 20 features per dm^2 and SIFT as the second best, averaging 5 features per dm^2.

Table 4.1: Number of Features detected for map image

Detector	Number of detected features
SURF	6192
SIFT	1525
ORB	500
BRISK	634
MSER	371

4.3 Test 2 : Number of Matches

This test consists of matching the map with an image used in the map's construction. The test uses the brute force matcher with k=2 and ratio test for outlier rejection. Ideally, the number of matches should coincide with the number of features, since both have the same illumination and proportions. Some differences may be accounted for by distortion and color equalization during the stitching.

The complete test of all 50 combinations ($5 detectors \times 5 extractors \times 2 matchers$) would prove impractical and unnecessary, since the tests's conditions are far worse than the idealized version explored here. For this reason any combination of detector/ extractor that does not produce enough matches will be discarded.

Table 4.2 shows the number of matches between map and inout image, also between parenteses the detected features for the input image. The table, color codes the best combinations with green and combinations that are not supported by the OpenCV library in red. The results for Flann-based search vary despite the constant input, for this reason the results were averaged for 20 attempts.

Table 4.2: Number of matches with semi-ideal conditions

Matcher	Detector	Extrator				
		SURF	SIFT	ORB	BRISK	FREAK
Brute force	SURF (223)	81	59	0	56	0
	SIFT (75)	2	15	-1	9	0
	ORB (31)	0	-1	0	0	0
	BRISK (3)	0	0	0	0	0
	MSER (5)	0	0	0	0	0
Flann-Based	SURF (223)	81,4	61,2	0	-1	-1
	SIFT (75)	2,01	15	-1	-1	-1
	ORB (31)	0	-1	0	-1	-1
	BRISK (3)	-1	-1	-1	-1	-1
	MSER (5)	-1	-1	-1	-1	-1

The best combinations are SURF/SURF and SURF/SIFT, matching respectively 30% and 25% of the features. A combination like SIFT/ SURF/ Brute-Force matched 2 out of 75 features less than 10%, for this reason it will not be included in the next stage of tests.

4.4 Test 3: Ratio

Because of the way that every extractor describes the feature, outlier rejection via ratio test requires different values of ratio. If ratio is set to 1, then every match passes the test and is considered an inlier. The smaller that value, the more strict the test becomes.

The value of ratio that maximizes inliers and minimizes outliers for each extractor was searched through trial and error and validated visually for different shades and rotations.

Table 4.3: Optimal ratio for match outlier rejection

Extractor	Optimal ratio
SURF	0.52
SIFT	0.72
BRISK	0.63

The extractors ORB and MSER failed to provide correct matches, and for this reason are not listed for this test. An extractor like SURF can operate using ratio=0.52, but the same ratio is too restrictive for the SIFT, which results in removing correct matches.

4.5 Test 4: Stills

This test consists of calculating the processing time, average number of matches, accuracy and precision for image inputs where there is no motion, hence stills.

The results for this test can be organized and filtered using several different parameters. On the one hand, one might compare the different combinations of detector/ extract and matcher, a second analysis can focus on which combination of outlier filters produces the best results. Another valid comparison would be to compare the bag's result to determine robustness to rotation and direct light. The complete version of the results is available in the appendix **??** with the results sorted by Filter/Bag/Matcher/Extractor/Detector.

4.5.1 Comparing Bags

The bags from figure 4.1, resulted in an attempt to choose inputs that are representative of rotation, position and light.

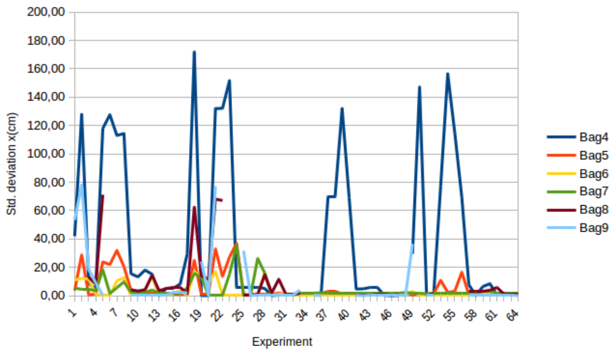
Green or Bag4: Rotation of 180^o and a light spot on its left down side.
Yellow or Bag5: Rotation of 135^o with no variation in illumination.
Black or Bag6: Rotation of 90^o and a small bright spot on the right down side.
Pink or Bag7: Rotation of -135^o and a sharp reflection introduced by illumination.
Blue or Bag8: No rotation and a slightly brighter texture, very close to ideal conditions.
Red or Bag9: Rotation of 90^o and an intense reflection on the right side of the image.

Figure 4.3a and 4.3b show the evolution of accuracy in the x axis, and θ respectively. Analyzing both images, bag 4 consistently underscores accuracy wise. A possible explanation is that the extractor cannot match features if the light varies, because it groups features according to a parameter that is susceptible to light.

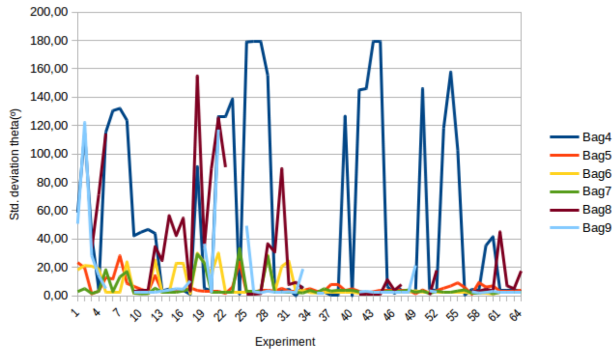
The rotation affected the number of matches negatively (figure 4.4a). Bag 8 outscored the other bags by at least 6 matches because of the lack of rotation and light variation. After bag 8, the best results came from bags 5 and 6, proving that light spots are more severe than rotation for the detection of features.

Figure 4.4b shows the relation between the bag and the processing time in the graph bag 4 is shown as having the longest cycle and bag 9 as having the shortest one. The highest difference between times is 150 ms, for SIFT/SIFT/FLANN and can be accounted for by the cover up of features due to light.

Table 4.4 lists the configuration for each experiment in the graphs. Table 4.5 shows a comparison between the bags. It was obtained by variating the bag while maintaining the parameters of the matching fixed (SURF/ SURF/ BF + Ratio). The table only contains a portion of the tests and as such may not be representative of the real precision/ accuracy, however the processing time and number of matches is correctly represented. The tests show processing time being smaller for bag 4, this happens because a light spot covers most of the image, thus obscuring a great amount of the features.

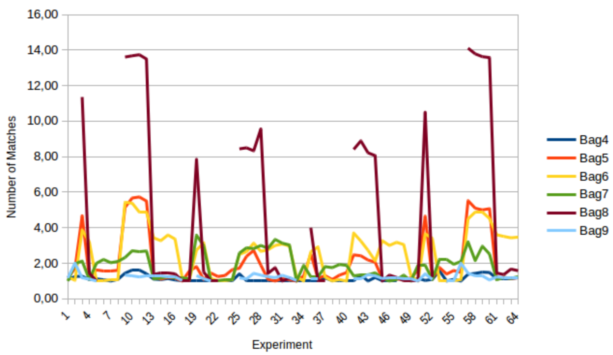


(a) Accuracy in the x axis vs Bags

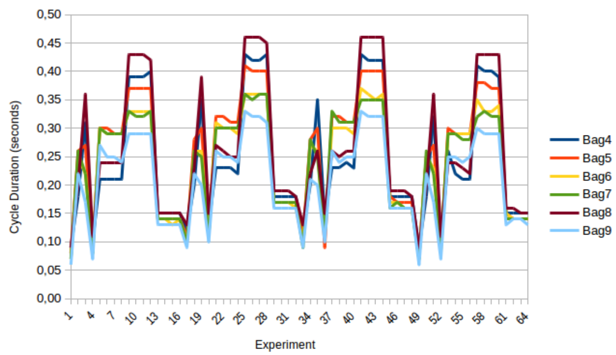


(b) Accuracy for theta vs Bags

Figure 4.3: Accuracy for x and theta vs Bags



(a) Number of matches vs bag



(b) Cycle Duration vs bag

Figure 4.4: Number of Matches and Cycle Duration vs Bags

Table 4.4: Experiments for Bag Comparison

Experience	Filter	Matcher	Detector	Extractor
1	Ratio	BF	SURF	BRISK
2	Ratio	BF	SIFT	SIFT
3	Ratio	BF	SURF	SIFT
4	Ratio	BF	SURF	SURF
5	Ratio	FB	SIFT	SIFT
6	Ratio	FB	SIFT	SIFT
7	Ratio	FB	SIFT	SIFT
8	Ratio	FB	SIFT	SIFT
9	Ratio	FB	SURF	SIFT
10	Ratio	FB	SURF	SIFT
11	Ratio	FB	SURF	SIFT
12	Ratio	FB	SURF	SIFT
13	Ratio	FB	SURF	SURF
14	Ratio	FB	SURF	SURF
15	Ratio	FB	SURF	SURF
16	Ratio	FB	SURF	SURF
17	Ratio + RANSAC	BF	SURF	BRISK
18	Ratio + RANSAC	BF	SIFT	SIFT
19	Ratio + RANSAC	BF	SURF	SIFT
20	Ratio + RANSAC	BF	SURF	SURF
21	Ratio + RANSAC	FB	SIFT	SIFT
22	Ratio + RANSAC	FB	SIFT	SIFT
23	Ratio + RANSAC	FB	SIFT	SIFT
24	Ratio + RANSAC	FB	SIFT	SIFT
25	Ratio + RANSAC	FB	SURF	SIFT
26	Ratio + RANSAC	FB	SURF	SIFT
27	Ratio + RANSAC	FB	SURF	SIFT
28	Ratio + RANSAC	FB	SURF	SIFT
29	Ratio + RANSAC	FB	SURF	SURF
30	Ratio + RANSAC	FB	SURF	SURF
31	Ratio + RANSAC	FB	SURF	SURF
32	Ratio + RANSAC	FB	SURF	SURF
33	Ratio + RANSAC + Coordinate Filter	BF	SURF	BRISK
34	Ratio + RANSAC + Coordinate Filter	BF	SIFT	SIFT
35	Ratio + RANSAC + Coordinate Filter	BF	SURF	SIFT
36	Ratio + RANSAC + Coordinate Filter	BF	SURF	SURF
37	Ratio + RANSAC + Coordinate Filter	FB	SIFT	SIFT
38	Ratio + RANSAC + Coordinate Filter	FB	SIFT	SIFT
39	Ratio + RANSAC + Coordinate Filter	FB	SIFT	SIFT
40	Ratio + RANSAC + Coordinate Filter	FB	SIFT	SIFT
41	Ratio + RANSAC + Coordinate Filter	FB	SURF	SIFT
42	Ratio + RANSAC + Coordinate Filter	FB	SURF	SIFT
43	Ratio + RANSAC + Coordinate Filter	FB	SURF	SIFT
44	Ratio + RANSAC + Coordinate Filter	FB	SURF	SIFT
45	Ratio + RANSAC + Coordinate Filter	FB	SURF	SURF
46	Ratio + RANSAC + Coordinate Filter	FB	SURF	SURF
47	Ratio + RANSAC + Coordinate Filter	FB	SURF	SURF
48	Ratio + RANSAC + Coordinate Filter	FB	SURF	SURF
49	Ratio + Coordinate Filter	BF	SURF	BRISK
50	Ratio + Coordinate Filter	BF	SIFT	SIFT
51	Ratio + Coordinate Filter	BF	SURF	SIFT
52	Ratio + Coordinate Filter	BF	SURF	SURF
53	Ratio + Coordinate Filter	FB	SIFT	SIFT
54	Ratio + Coordinate Filter	FB	SIFT	SIFT
55	Ratio + Coordinate Filter	FB	SIFT	SIFT
56	Ratio + Coordinate Filter	FB	SIFT	SIFT
57	Ratio + Coordinate Filter	FB	SURF	SIFT
58	Ratio + Coordinate Filter	FB	SURF	SIFT
59	Ratio + Coordinate Filter	FB	SURF	SIFT
60	Ratio + Coordinate Filter	FB	SURF	SIFT
61	Ratio + Coordinate Filter	FB	SURF	SURF
62	Ratio + Coordinate Filter	FB	SURF	SURF
63	Ratio + Coordinate Filter	FB	SURF	SURF
64	Ratio + Coordinate Filter	FB	SURF	SURF

Table 4.5: Results for test 4 using Experiment 4

SURF/ SIFT/ BFMatcher + Ratio	Processed Images	Localization calculated	Localization not filtered	Localization correctly calculated	Accuracy Test			Precision test			Average processing time(ms)	Average matches per image
					Average Error x (cm)	Average Error y (cm)	Average Error theta (º)	Standard Deviation x(cm)	Standard Deviation y(cm)	Standard Deviation theta(º)		
Bag4	90	77	77	76	-0,44	0,86	-3,33	4,62	5,35	3,12	0,10	1,08
Bag5	99	51	51	51	-0,92	0,41	0,40	0,91	0,64	3,31	0,09	1,22
Bag6	64	63	63	62	0,27	0,04	4,74	1,20	0,35	18,43	0,09	3,21
Bag7	82	56	56	54	-2,11	-1,10	0,80	2,94	2,09	3,11	0,08	1,09
Bag8	59	36	36	21	3,09	1,46	-21,90	7,67	8,26	67,92	0,11	1,44
Bag9	103	71	71	69	0,90	-0,01	0,30	9,12	6,52	14,32	0,07	1,11

4.5.2 Comparing Matcher Combinations

The combinations of detector, extractor and matcher tested are:

SURF/ SURF/ Brute-Force
SURF/ SIFT/ Brute-Force
SURF/ BRISK/ Brute-Force
SIFT/ SIFT/ Brute-Force
SURF/ SURF/ Flann-Based
SURF/ SIFT/ Flann-Based
SIFT/ SIFT/ Flann-Based

Every test for the Flann based search was repeated 4 times and averaged to produce a more representative result.

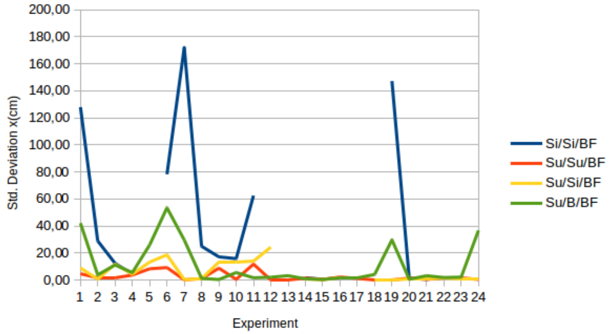
The brisk extractor was the least reliable option. Figure 4.7a shows the percentage of times each detector/extractor combination produces coordinates. Most of the times the combination SIFT/SIFT and SURF/BRISK outputs coordinates for only 20% of the input image, much lower than 50% for SURF/SURF or SURF/SIFT. Figure 4.7b shows the percentage of times each detector/extractor combination produces correct coordinates. Only the SURF/SURF combination produces correct coordinates at least 50% of the times, the other combinations require the addition of filters to display results with the same accuracy.

For bag 9 that shows high variation in illumination, SIFT/ SIFT is the least accurate combination, followed by SURF/ SIFT whose orientation lacks accuracy.

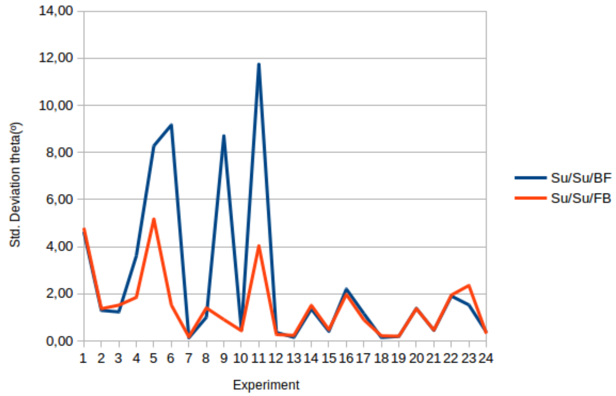
SURF/ SURF consistently outperforms all other combinations with a standard deviation bellow 10^{o} and 5 cm even in the worst conditions (figures 4.5a, 4.5b, 4.6a, 4.6b). The combination that most consistently delivers correct coordinates is SURF/ SURF. If the conditions are ideal (bag 4), approximately 90% of the time the calculated coordinate is the real coordinate, the value drops to 40% when RANSAC + Ratio was used as filter.

The combination SURF/ SIFT consistently produces more matches than the rest, with up to 14 matches (figure 4.8b). The second best combination manages to produce 3 matches for the same input. Even though, more matches decrease error by averaging deviations, the results showed that SURF/SURF is the most accurate combination. Table 4.6 shows the combinations used for each experiment in the graphs.

Table 4.7 was produced by changing the detector/ extractor/ matcher combination while maintaining the filter and bag. Duration wise, the fastest combinations are SURF/BRISK or SURF/ SURF, averaging 100ms per cycle, (see figure 4.8a) the slowest option is SURF/SIFT at 300ms. The Flann-Based matcher is the slowest matcher, causing a delay of 50ms for the SURF/SURF and 100ms for SURF/ SIFT. This increase in time however results in improved precision for the combination SURF/ SURF (see figure 4.5b).

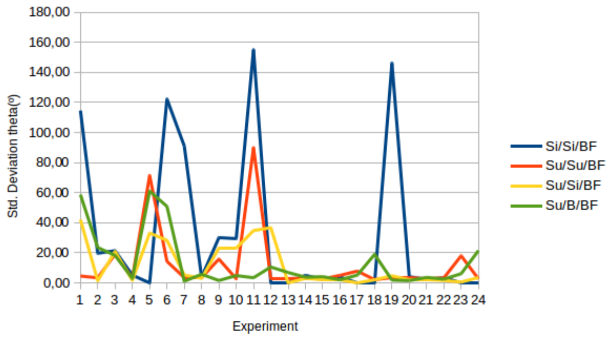


(a) Accuracy in x axis vs Detector/Extractor

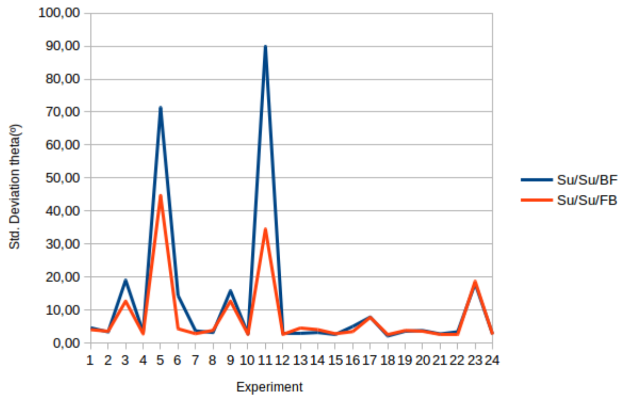


(b) Accuracy in x axis vs Matchers

Figure 4.5: Accuracy in x axis vs Detector/Extractor/Matcher

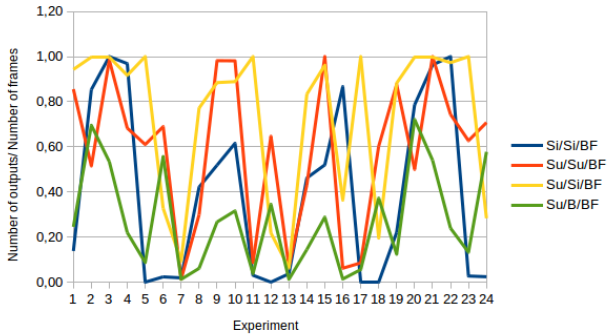


(a) Accuracy in theta vs Detector/Extractor

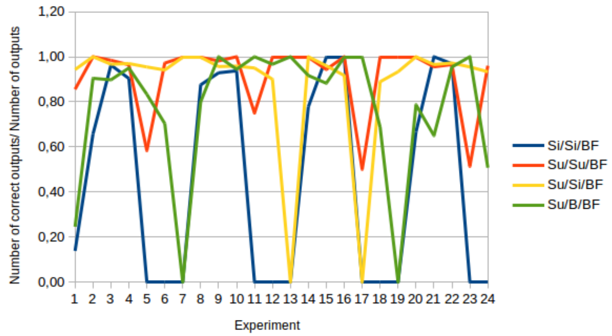


(b) Accuracy in theta vs Matcher

Figure 4.6: Accuracy in theta vs Detector/Extractor/Matcher

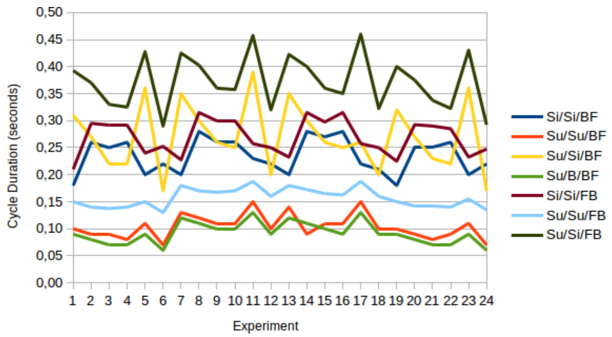


(a) Outputs Ratio vs Detector/Extractor/Matcher

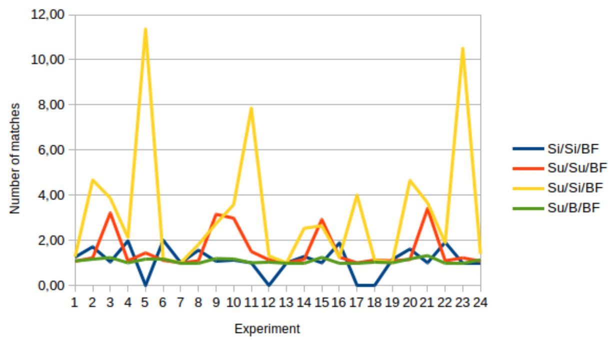


(b) Correct Output Ratio vs Detector/Extractor/Matcher

Figure 4.7: Output Ratio vs Detector/ Extractor/ Matcher



(a) Cycle Duration vs Detector/Extractor/Matcher



(b) Number of Matches vs Detector/Extractor/Matcher

Figure 4.8: Cycle Duration and Number of Matches vs Detector/Extractor/Matcher

Table 4.6: Experiments for Detector/Extractor/Matcher Comparison

Experience	Filter	Bag	Experience	Filter	Bag
1	Ratio	4	13	Ratio + RANSAC + Coordinate Filter	4
2		5	14		5
3		6	15		6
4		7	16		7
5		8	17		8
6		9	18		9
7	Ratio + RANSAC	4	19	Ratio + Coordinate Filter	4
8		5	20		5
9		6	21		6
10		7	22		7
11		8	23		8
12		9	24		9

Table 4.7: Results for test 4 using Experiment 2

Bag 5 + Ratio Test							Accuracy Test			Precision test				
Matcher	Detetor	Extrator	Processed Images	Localization calculated	Localization not filtered	Localization correctly calculated	Average Error x (cm)	Average Error y (cm)	Average Error theta (º)	Standard Deviation x(cm)	Standard Deviation y(cm)	Standard Deviation theta(º)	Average processing time(ms)	Average matches per image
Brute Force	SURF	SURF	99	51	51	51	-0,55	0,30	0,40	0,54	0,47	3,31	0,09	1,22
		SIFT	39	39	39	39	-0,39	-0,05	-0,82	0,14	0,14	1,12	0,27	4,67
		BRISK	105	73	73	66	-0,58	1,69	-6,93	2,17	4,11	22,44	0,08	1,16
	SIFT	SIFT	41	35	35	23	-3,39	-7,72	-4,76	16,88	16,19	19,12	0,26	1,71
Flann Based	SURF	SURF	69	42	42	42	-0,60	0,27	0,67	0,52	0,52	3,39	0,14	1,21
			68	31	31	31	-0,61	0,37	0,51	0,48	0,56	3,41	0,14	1,23
			68	35	35	35	-0,63	0,35	0,42	0,49	0,54	3,36	0,14	1,11
			69	35	35	35	-0,73	0,30	1,29	0,53	0,45	3,52	0,14	1,17
		SIFT	30	30	30	29	-0,33	0,06	-0,63	0,98	0,83	2,90	0,37	5,73
			30	30	30	26	-0,66	0,69	-5,14	1,33	2,64	13,24	0,37	5,50
			30	30	30	29	-0,54	0,10	-1,03	0,41	1,00	4,52	0,37	5,67
			29	29	29	27	-0,18	-0,05	-1,25	2,39	2,00	6,53	0,37	5,14
	SIFT	SIFT	36	26	26	19	-6,19	-8,06	-2,20	11,61	14,78	11,68	0,30	1,54
			35	29	29	21	-6,40	-6,02	-4,51	12,72	12,76	12,04	0,30	1,62
			38	30	30	21	-5,02	-4,92	-1,80	11,23	10,90	8,59	0,29	1,60
			37	31	31	20	-3,63	-8,01	0,84	18,72	15,54	28,24	0,29	1,55

4.5.3 Comparing Filters Combinations

The combinations of coordinate filters and match filters tested are:

Ratio Test or **R** for short.

Ratio and RANSAC Test or **RaR** for short.

Ratio and RANSAC Test and Coordinate Rejection Filter or **RaRRe** for short.

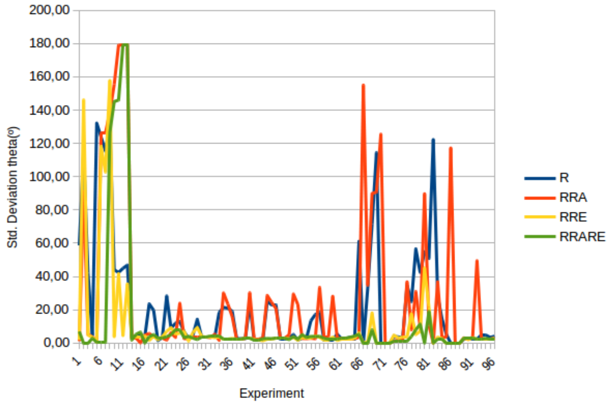
Ratio Test and Coordinate Rejection Filter or **RRe** for short.

The tests do not show direct correlation between the processing time and the filter combination, see figure 4.12a. In some cases the fastest combination is RRa, other times RRe. One possible explanation has to do with the processing cycle, if no matches remain after the RANSAC filter, the cycle shortcuts and does not calculate the coordinate. On the other hand if the coordinate is filtered by the RRe, time is saved by not calculating statistics.

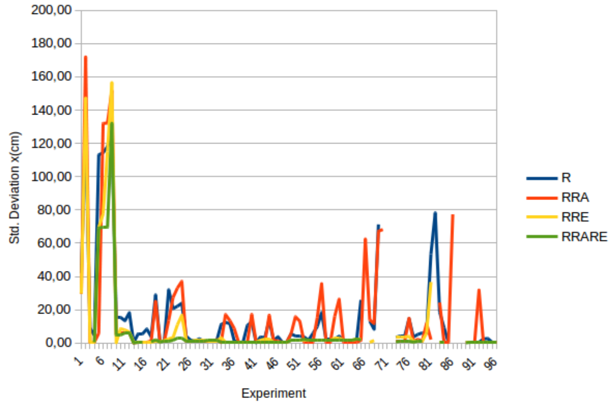
Figure 4.11a shows the output ratio given by number of outputs over number of input images, while figure 4.11b shows the correct output ratio given by number of correct outputs over the number of outputs, in both graphs the more filters are used, the less outputs are produced. Table 4.9 clearly shows this drop in outputs and the consequent improvement in accuracy and precision. Precision and accuracy wise, RaRRe was the best filter combination, followed by the RRe and finally RaR (see figures 4.12a, 4.9a, 4.10a, 4.10b).

Table 4.9 was produced by varying the filter combination while maintaining the same input and detector/ extractor/ matcher combination. Table 4.9 shows a comparison between filters using the combination SIFT/ SIFT/ Brute-Force for bag 4, the best filter being RRe. By adding RANSAC a great amount of correct coordinates were filtered, Not all detector/extractor/ matcher combinations have this vulnerability, but this shows that more filters does not necessarily mean better results.

Table 4.8 describes the combinations used for each experiment in the graphs.

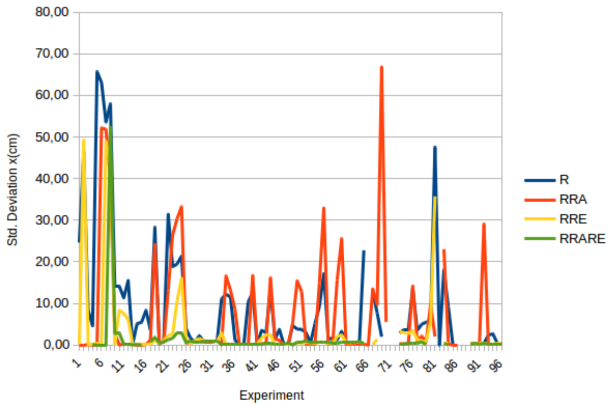


(a) Accuracy in theta vs Filter

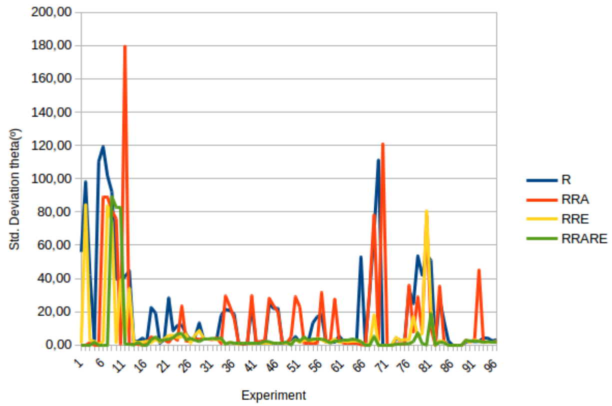


(b) Accuracy for x axis vs Filter

Figure 4.9: Accuracy vs Filter

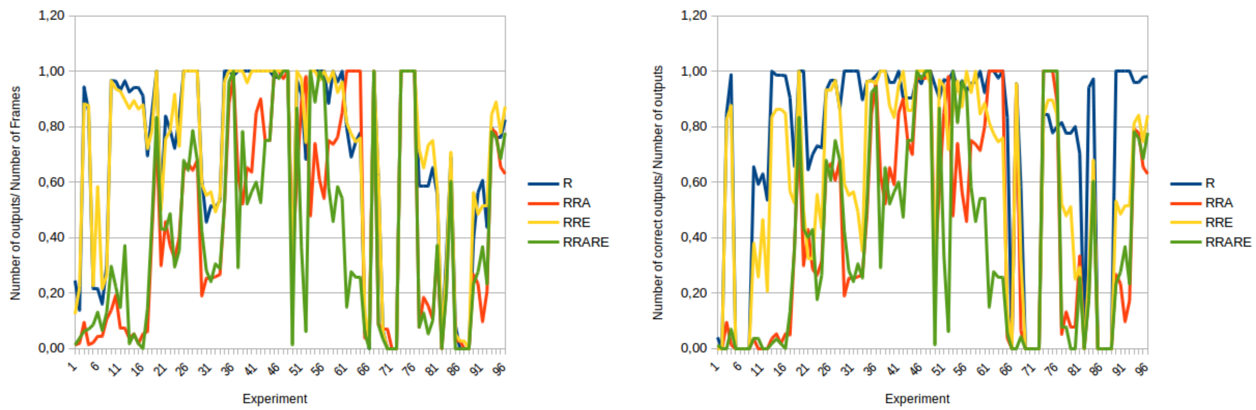


(a) Precision for x axis vs Filter



(b) Precision in theta vs Filter

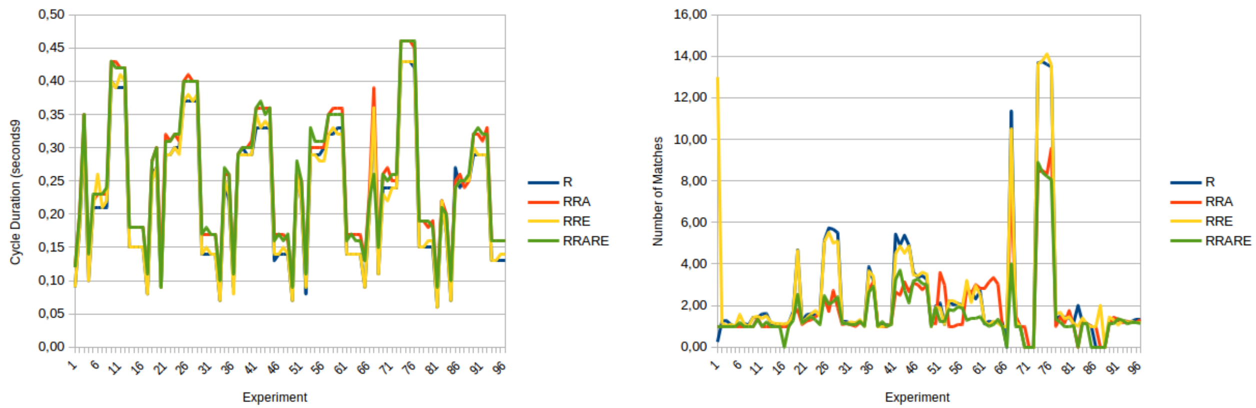
Figure 4.10: Precision vs Filter



(a) Output Ratio vs Filter

(b) Correct Output Ratio vs Filter

Figure 4.11: Output Ratio vs Filter



(a) Cycle Duration vs Filter

(b) Number of Matches in theta vs Filter

Figure 4.12: Cycle Duration and Number of Matches vs Filter

Table 4.8: Experiments for Filter Comparison

Experience	Bag	Matcher	Detector	Extractor
1	4	BF	SURF	BRISK
2			SIFT	SIFT
3			SURF	SIFT
4				SURF
5		FB	SIFT	SIFT
6				
7				
8				
9			SURF	SIFT
10				
11				
12				
13				SURF
14				
15				
16				
17	5	BF	SURF	BRISK
18			SIFT	SIFT
19			SURF	SIFT
20				SURF
21		FB	SIFT	SIFT
22				
23				
24				
25			SURF	SIFT
26				
27				
28				
29				SURF
30				
31				
32				
33	6	BF	SURF	BRISK
34			SIFT	SIFT
35			SURF	SIFT
36				SURF
37		FB	SIFT	SIFT
38				
39				
40				
41			SURF	SIFT
42				
43				
44				
45				SURF
46				
47				
48				
49	7	BF	SURF	BRISK
50			SIFT	SIFT
51			SURF	SIFT
52				SURF
53		FB	SIFT	SIFT
54				
55				
56				
57			SURF	SIFT
58				
59				
60				
61				SURF
62				
63				
64				
65	8	BF	SURF	BRISK
66			SIFT	SIFT
67			SURF	SIFT
68				SURF
69		FB	SIFT	SIFT
70				
71				
72				
73			SURF	SIFT
74				
75				
76				
77				SURF
78				
79				
80				
81	9	BF	SURF	BRISK
82			SIFT	SIFT
83			SURF	SIFT
84				SURF
85		FB	SIFT	SIFT
86				
87				
88				
89			SURF	SIFT
90				
91				
92				
93				SURF
94				
95				
96				

Table 4.9: Results for test 4 using Experiment 3

SURF/ SIFT/ BFMatcher + Bag4					Accuracy Test			Precision test				
	Processed Images	Localization calculated	Localization not filtered	Localization correctly calculated	Average Error x (cm)	Average Error y (cm)	Average Error theta (º)	Standard Deviation x(cm)	Standard Deviation y(cm)	Standard Deviation theta(º)	Average processing time(ms)	Average matches per image
R	35	33	33	28	-2,89	9,64	-0,47	8,37	23,66	42,09	0,18	1,27
Rre	34	30	28	28	0,14	0,25	-3,75	0,24	0,12	2,84	0,19	1,03
RaR	32	3	3	3	0,25	0,29	-5,04	0,11	0,06	1,37	0,21	1,00
RaRRe	32	2	0	0	-nan	-nan	-nan	-nan	-nan	-nan	0,21	1,00

Chapter 5

Conclusion

This work describes the design and test of a localization system called "FeatLoc", usable for robotic applications. It is made up of two subsystems, one using visual odometry and the other visual memory, that is, it searches a global map for known features. All the code was developed using the ROS framework and OpenCV. The visual odometry produces a relative coordinate by comparing consecutive input images from the camera while the visual memory produces an absolute coordinate by comparing the input to the global map.

The localization system is designed to operate using features of natural origin, thus providing an alternative solution when artificial beacons are not usable or existent. The reason why natural features are appealing for the FeatLoc is because there is abundance and variety of features, a floor made of natural wood or a granite has very distinctive patterns. Variety of features means uniqueness or distinctiveness in characteristics, either shape or contrast, ideally the more distinctive the features the easier it becomes to use the localization system. On the other hand, the artificial vision systems that acquire the input image have to be able to capture the differences between tiles or boards, in other words the cameras need enough resolution and sensitivity to contrast to be able to tell apart the locations

The test setup is made up of camera mounted Robot, moving on an indoor scenario of about 2m x 1.5m made of tiles from floating wood. The tiles were installed without repeating tiles to resemble the shape of a rectangle. The tiles were kept clean throughout the tests to prevent covering features. The robot is an adapted RC car platform with Ackerman steering controlled by an Arduino and commanded by a PC. The PC has an i3 processor, runs Linux 16.04 and is connected to a 620x480 pixel camera. The Arduino runs a ROS node that receives the commands from a Wiimote and control the motors through the production of PWM signals. All the code for this system was written using OpenCV and ROS. The camera was mounted so that the image pixel size is 0.8mm or above. A homography node was implemented to correct the distortion introduced by the camera lens and to make the input image resemble an image filmed from above. The global map is created using images after the camera calibration and homography, resulting in a map resolution of 0.8mm. The construction of the map required 50 individual images and the use of Panoramic Stitcher software (Hugin - Panorama photo stitcher), although the stitched map has stickers, after the map construction, all sticker were removed.

Localization is achieved by matching features from the global map to features from the real time image retrieved by the camera. This matching is made using both map and input image after the homography. The feature correspondence process is made up of 3 phases: detection, extraction and matching. The system is able to detect features of the following types: ridges, blobs, edges and corners. Tested detectors are SURF, SIFT, ORB, BRISK and MSER. Tested extractors are: SURF, SIFT, BRISK, ORB and FREAK. Used matchers are: Brute Force + KNN, FLANN + KNN. Posterior validation of the matches is achieved by ratio test and RANSAC.

In parallel with the previous approach, a simple visual odometry subsystem was developed to complement the output with relative coordinate information. This subsystem uses the same feature matching architecture (but matching consecutive frames of the camera). Both absolute and relative coordinates are fused by a node, that validates the output using a filter that discards coordinates when the variation is above a certain threshold (20cm or 10^{o}).

This localization system can be used in many applications, for instance, service robots and can be easily reproduced using somewhat cheap hardware (webcam, remote controlled car, wiimote).

The system was run on a laptop with the specifications:

Linux 16.04
ROS Indigo
Processor: Intel Core i3 CPU M 370 2.40GHz x 4
RAM: 4Gb

5.1 Summary of Results

The fastest extractors are not necessarily the best. BRISK may operate at a faster rate of 60ms/ cycle, but most processed images don't produce coordinates, contrary to a 80ms cycle for SURF with 50% of input images generating coordinates.

The use of FLANN-based matcher improves accuracy the SURF detector, but it adds 30ms to the processing time.

Using SURF/SURF/BF for the worst bag, the Ratio Test + RANSAC Test + Coordinate Filter produced a max error under 4 cm and 3^{o}, Ratio Test + RANSAC Test got an error bellow 16 cm and 90^{o} And Ratio Test + Coordinate Filter returned an error below 4cm and 18^{o}.

The best matcher combination proved to be SURF/ SURF/ FLANN with Ratio Test + RANSAC Test + Coordinate Filter. It has the best accuracy and precision. However the combination tends to filter most of its outputs, ending up with only an average of 2 outputs for 50 input images.

Due to this limitation, a more stable application would use SURF/ SURF/ FLANN with Ratio Test + Coordinate Filter filters. The coordinate output rate is higher with up to 10 correct localizations per second, and below 3cm 3^{o} if the input image does not have ideal conditions illumination and rotation wise. A high output rate means that a robot using this system does not remain lost for a long time. If the features are not obscured, the system can produce a new coordinate every

100ms, a typical processing cycle includes receiving an input image, detecting, extracting and matching features and finally calculating and filtering the coordinates.

Finally the results proved that the system is more robust to rotation, than to variation in light. The best results come from bags 5, 6, 9 which show a great amount of detail and the correct type of shade, while the worst results come from 8 where the input has a slightly brighter texture.

5.1.1 Industry Adoption

A more affordable/ industrialized version of this system could run on Raspbery PI. This version would maintain resolution but with a reduced rate of coordinate outputs.

A possible applications for this system is the automatic storage industry. This system competes with line followers that have very cheap sensors but somewhat expensive setup. The setup for this system is very time consuming but does not require infrastructure cost. It only requires that the floor has a non-repetitive pattern and a camera with resolution to capture the texture.

The construction of the map is the greatest disadvantage of this system. A 2x1.5m map like the one used in this dissertation required 50 images to cover the entire area, all taken with very specific rotation and light. The capture took 4 hours and several attempts, added to the 3 hours for the map's stitching.

5.2 Admitted Limitations

This method is heavily conditioned by the resolution of the input image. An increase in resolution equates to an increase in details and features that produce more matches.

The real world dimensions of the map is 138x217cm, mapped on a 1761x2810 px image. The resulting resolution is 0.8 mm/pixel for the x and y axes. The feed image is mapped on 458x505px with real world dimensions 36x39cm, the resolution is approximately the same as the maps (see Table 5.1).

The localization system is limited by the amount of detail and features that can be detected on the input image. The best combination of filters and coordinates (RRe SURF/SURF/BF) achieved 2cm resolution, only 1cm away from the theoretical limit for the resolution.

Table 5.1: Resolution for map and input image

	Map		Input	
Unit	Width	Height	Width	Height
mm	1761	2810	458	505
pixel	1380	2170	360	390
Resolution (mm/px)	0,78	0,77	0,79	0,77

Coordinates are as good as the stitched map. As such, a second limitation is the quality of the stitched map. Any distortion added to the map causes systematic errors to be added to the output coordinates.

The FeatLoc system was tested on a board containing repeated boards, even though the ratio test is designed to resist this situation of multiple good matches. The results show that if the repeated boards are not represented with the same exact shade, there is a chance that the matches may get mixed up resulting in incorrect coordinates.

5.3 Future Work

5.3.1 System optimization

In order to operate in a 25frame/ second rate, the computation cycle can not exceed 40ms. For this reason, the system requires some optimization. A possibility is to divide the map in sections and search in smaller areas according to the previous coordinates.

Also, the construction of the map requires optimization. A possible solution would be to use a table-like structure with a camera facing down to capture input images. This combined with a more robust stitching software could reduce the setup time to minutes.

Finally an increase in robustness to light would prove useful. This can be achieved by two different adaptations. The first and less expensive is to use a veil or umbrella to create a more uniform illumination, the second alternative consists of a polarizing filter installed after the camera lens to suppress glare from the surface.

Appendix A

ROS

This appendix can be skipped if the reader is familiar with the ROS platform.

ROS, standing for Robotic Operating System, is a framework for the development of robotic software.

To understand the ROS framework the concept of a node, topic and bag have be mastered.

A node is a processing unit, it is a program that runs on the computer and performs a certain function. A computer can run more than one node at the same time, and even several instances of the same node as long as different names are assigned for each.

A topic is a communication channel used for message exchange. ROS supports the subscriber/ publisher model, in which a node can write/ publish a message in a topic and a second node can read/ subscribe to that topic, receiving messages published there. This functionality means that the communications of variables is managed automatically, facilitating the task of the programmer. A topic can have more than one publisher, more than one subscriber and the message can have any type (int, float, string, matrix).

A bag is a recording of messages from a topic, the ROS platform allows the user to replay a sequence of messages in the same conditions that they were acquired. This functionality allows the programmer to have repeatable results, that is, different combinations of parameters can be tested for the same exact input.

Appendix B

Results

This results relate to the tests using image sequences without movement. The coordinates are calculated using only the visual memory node (testar). The results are organized by Filter/ Bag/ /Matcher/Extractor/Detector.

Ratio

								Accuracy Test						Precision test				
	Matcher	Detetor	Extrator	Processed Images	Localization calculated	Localization not filtered	Localization calculated correctly	Standard Deviation x(cm)	Standard Deviation y(cm)	Standard Deviation theta(º)	Average Error x (cm)	Average Error y (cm)	Average Error theta (º)	Standard Deviation x(cm)	Standard Deviation y(cm)	Standard Deviation theta(º)	Average processing time(sec)	Average matches per image
Bag4	Brute Force	SURF	SURF	90	77	77	76	4,63	5,42	4,56	-0,44	0,86	-3,33	4,62	5,35	3,12	0,10	1,08
			SIFT	35	33	33	28	8,85	25,55	42,09	-2,89	9,64	-0,47	8,37	23,66	42,09	0,31	1,27
			BRISK	102	25	25	1	42,04	65,41	58,66	-34,07	58,60	17,79	24,62	29,06	55,90	0,09	1,08
		SIFT	SIFT	58	8	8	0	127,88	3,86	114,58	-119,17	2,97	-59,25	46,39	2,47	98,07	0,18	1,25
	Flann Based	SURF	SURF	67	63	63	62	5,46	2,74	4,76	-0,62	0,57	-2,45	5,43	2,69	4,08	0,15	1,08
				67	62	62	62	0,20	0,27	3,59	0,07	0,24	-2,74	0,19	0,13	2,32	0,15	1,11
				68	64	64	63	5,15	1,49	3,99	-0,52	0,44	-3,37	5,13	1,42	2,12	0,15	1,14
				68	62	62	61	8,38	2,39	3,69	-0,98	0,53	-3,02	8,32	2,33	2,12	0,15	1,06
			SIFT	29	28	28	15	18,17	32,85	46,79	-9,51	20,92	14,35	15,49	25,33	44,54	0,39	1,61
				28	27	27	16	15,62	31,63	42,37	-6,52	18,59	18,02	14,20	25,58	38,34	0,39	1,44
				30	29	29	19	15,21	35,17	43,98	-5,31	20,19	18,73	14,25	28,80	39,79	0,40	1,41
				29	27	27	17	13,36	29,84	44,86	-7,08	17,75	17,90	11,34	24,00	41,13	0,39	1,59
		SIFT	SIFT	51	11	11	0	114,39	37,25	123,74	-95,41	19,23	-7,94	63,10	31,90	119,28	0,21	1,09
				51	11	11	0	113,08	42,99	132,15	-92,01	23,74	-32,61	65,72	35,84	110,28	0,21	1,00
				50	8	8	0	118,01	19,15	115,31	-105,13	8,11	-54,35	53,61	17,35	101,70	0,21	1,12
				50	15	15	0	127,67	21,91	130,52	-113,74	7,32	-71,63	57,99	20,65	92,10	0,21	1,07
Bag5	Brute Force	SURF	SURF	99	51	51	51	1,29	0,77	3,33	-0,92	0,41	0,40	0,91	0,64	3,31	0,09	1,22
			SIFT	39	39	39	39	0,70	0,19	1,39	-0,66	-0,07	-0,82	0,24	0,18	1,12	0,27	4,67
			BRISK	105	73	73	66	3,77	6,05	23,49	-0,97	2,30	-6,93	3,64	5,59	22,44	0,08	1,16
		SIFT	SIFT	41	35	35	23	28,87	24,37	19,70	-5,68	-10,49	-4,76	28,30	22,00	19,12	0,26	1,71
	Flann Based	SURF	SURF	69	42	42	42	1,32	0,79	3,46	-1,00	0,36	0,67	0,86	0,71	3,39	0,14	1,21
				68	31	31	31	1,30	0,92	3,45	-1,02	0,51	0,51	0,80	0,77	3,41	0,14	1,23
				68	35	35	35	1,33	0,87	3,39	-1,05	0,48	0,42	0,82	0,73	3,36	0,14	1,11
				69	35	35	35	1,51	0,74	3,75	-1,22	0,40	1,29	0,89	0,62	3,52	0,14	1,17
			SIFT	30	30	30	29	1,73	1,13	2,96	-0,55	0,09	-0,63	1,64	1,13	2,90	0,37	5,73
				30	30	30	26	2,49	3,71	14,20	-1,11	0,94	-5,14	2,22	3,58	13,24	0,37	5,50
				30	30	30	29	1,13	1,36	4,64	-0,90	0,13	-1,03	0,68	1,35	4,52	0,37	5,67
				29	29	29	27	4,03	2,72	6,65	-0,31	-0,07	-1,25	4,01	2,72	6,53	0,37	5,14
		SIFT	SIFT	36	26	26	19	22,07	22,87	11,89	-10,39	-10,95	-2,20	19,47	20,08	11,68	0,30	1,54
				35	29	29	21	23,88	19,18	12,86	-10,73	-8,18	-4,51	21,32	17,34	12,04	0,30	1,62
				38	30	30	21	20,63	16,25	8,78	-8,43	-6,69	-1,80	18,83	14,81	8,59	0,29	1,60
				37	31	31	20	31,97	23,77	28,25	-6,09	-10,88	0,84	31,39	21,13	28,24	0,29	1,55
Bag6	Brute Force	SURF	SURF	64	63	63	62	1,23	0,35	19,03	0,27	0,04	4,74	1,20	0,35	18,43	0,09	3,21
			SIFT	30	30	30	29	11,69	4,85	20,91	-1,94	0,34	-2,21	11,53	4,83	20,80	0,22	3,87
			BRISK	73	39	39	35	11,10	6,21	18,27	-0,32	0,39	-4,16	11,09	6,20	17,79	0,07	1,23
		SIFT	SIFT	27	27	27	26	12,34	5,18	21,39	-1,87	1,55	-1,84	12,19	4,94	21,31	0,25	1,04
	Flann Based	SURF	SURF	46	45	45	43	3,74	0,36	22,63	-0,32	0,01	5,43	3,73	0,36	21,96	0,14	3,36
				44	44	44	44	0,44	0,35	2,58	0,38	-0,04	2,31	0,22	0,34	1,14	0,14	3,43
				45	45	45	45	0,45	0,26	2,61	0,41	0,04	2,47	0,18	0,26	0,86	0,14	3,27
				45	44	44	43	1,43	0,26	22,72	0,20	0,04	6,00	1,42	0,26	21,92	0,13	3,59
			SIFT	21	21	21	19	3,15	2,08	1,71	-1,07	0,13	1,35	2,96	2,07	1,04	0,33	5,38
				21	21	21	21	0,17	0,76	1,72	0,11	-0,71	1,51	0,12	0,26	0,82	0,33	5,43
				21	21	21	19	3,60	1,67	1,85	-0,87	-0,12	1,56	3,49	1,66	0,99	0,33	4,90
				21	21	21	19	13,98	6,05	25,19	-3,27	0,88	-4,25	13,58	5,99	24,83	0,33	4,90
		SIFT	SIFT	23	23	23	23	0,60	0,69	2,58	0,57	0,63	2,50	0,17	0,29	0,65	0,30	1,04
				24	24	24	24	0,61	0,71	2,63	0,57	0,61	2,45	0,24	0,36	0,96	0,29	1,04
				24	24	24	23	10,53	2,19	2,67	-1,57	1,04	2,42	10,41	1,92	1,13	0,29	1,08
				24	24	24	23	12,76	5,41	23,93	-2,08	1,66	-2,58	12,58	5,15	23,79	0,29	1,08

	Matcher	Detetor	Extrator	Processed Images	Localization calculated	Localization not filtered	Localization calculated correctly	Accuracy Test						Precision test			Average processing time(sec)	Average matches per image
								Standard Deviation x(cm)	Standard Deviation y(cm)	Standard Deviation theta(º)	Average Error x (cm)	Average Error y (cm)	Average Error theta (º)	Standard Deviation x(cm)	Standard Deviation y(cm)	Standard Deviation theta(º)		
Bag7	Brute Force	SURF	SURF	82	56	56	54	3,61	2,36	3,21	-2,11	-1,10	0,80	2,94	2,09	3,11	0,08	1,09
			SIFT	36	33	33	32	4,30	2,75	1,66	-2,09	-1,10	0,11	3,76	2,52	1,66	0,22	2,12
			BRISK	91	20	20	19	5,49	3,50	2,96	-2,89	-1,60	2,57	4,66	3,11	1,47	0,07	1,00
		SIFT	SIFT	32	31	31	28	4,29	3,30	5,10	-1,99	-0,14	-0,72	3,81	3,30	5,05	0,26	1,97
	Flann Based	SURF	SURF	55	41	41	40	1,92	1,41	3,48	-1,76	-0,82	0,81	0,76	1,14	3,38	0,14	1,17
				54	42	42	42	1,88	1,04	2,30	-1,81	-0,99	1,13	0,54	0,33	2,00	0,14	1,17
				55	38	38	38	1,87	1,04	2,69	-1,75	-0,82	0,91	0,64	0,64	2,53	0,14	1,24
				54	43	43	43	1,71	1,05	2,50	-1,60	-0,72	0,29	0,61	0,77	2,48	0,14	1,12
			SIFT	26	26	26	25	2,41	1,04	1,55	-1,79	-0,62	0,68	1,61	0,84	1,40	0,32	2,65
				26	25	25	25	2,21	1,02	1,86	-1,82	-0,71	0,74	1,23	0,74	1,70	0,33	2,32
				26	23	23	22	2,27	1,11	1,45	-1,75	-0,54	0,45	1,44	0,97	1,38	0,32	2,70
				26	26	26	24	4,07	4,30	5,36	-2,41	-1,05	-0,63	3,28	4,18	5,33	0,33	2,69
		SIFT	SIFT	29	29	29	29	1,69	1,08	2,95	-1,63	-0,51	1,41	0,48	0,96	2,60	0,29	2,21
				29	29	29	27	5,72	2,63	13,36	-2,55	-0,76	-1,35	5,12	2,52	13,29	0,29	2,03
				29	29	29	28	9,86	2,65	17,09	-3,37	-0,93	-1,80	9,26	2,49	17,00	0,29	2,10
				28	28	28	26	18,29	12,41	18,35	-6,42	-1,53	-1,55	17,13	12,32	18,28	0,30	2,00
Bag8	Brute Force	SURF	SURF	59	36	36	21	8,28	8,38	71,36	3,09	1,46	-21,90	7,67	8,26	67,92	0,11	1,44
			SIFT	22	22	22	21	13,21	1,80	33,15	3,14	-0,47	6,94	12,83	1,74	32,42	0,36	11,36
			BRISK	69	6	6	5	25,89	4,05	61,04	12,35	-3,50	30,62	22,76	2,03	52,81	0,09	1,17
		SIFT	SIFT	37	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,20	-nan
	Flann Based	SURF	SURF	46	27	27	22	3,64	3,04	24,89	1,03	-0,33	1,02	3,49	3,02	24,87	0,15	1,37
				46	27	27	21	5,13	4,98	56,54	1,18	-0,49	-18,43	4,99	4,96	53,46	0,15	1,48
				46	30	30	24	6,14	6,40	54,81	2,05	0,36	-7,00	5,78	6,39	54,36	0,15	1,40
				46	27	27	21	5,77	5,39	42,46	1,83	0,61	-5,92	5,48	5,36	42,04	0,15	1,44
			SIFT	18	18	18	14	4,31	2,60	3,15	2,37	1,20	1,31	3,60	2,31	2,87	0,43	13,61
				19	19	19	16	4,31	2,95	3,65	2,36	1,07	1,99	3,61	2,75	3,06	0,43	13,74
				19	19	19	16	3,41	2,58	3,92	1,79	0,76	0,83	2,90	2,47	3,83	0,43	13,68
				20	20	20	16	14,59	3,16	34,66	5,11	0,29	8,96	13,67	3,15	33,49	0,42	13,50
		SIFT	SIFT	32	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,24	-nan
				31	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,24	-nan
				31	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,24	-nan
				31	2	2	0	71,26	5,86	114,39	71,23	-2,50	27,30	1,99	5,30	111,09	0,24	1,00
Bag9	Brute Force	SURF	SURF	103	71	71	69	9,17	6,52	14,32	0,90	-0,01	0,30	9,12	6,52	14,32	0,07	1,11
			SIFT	52	17	17	16	18,60	8,60	28,26	4,57	1,77	-5,69	18,03	8,42	27,68	0,17	1,18
			BRISK	115	64	64	45	53,41	45,77	50,82	24,31	21,91	2,45	47,56	40,18	50,76	0,06	1,17
		SIFT	SIFT	43	1	1	0	78,20	36,81	122,23	78,20	36,81	-122,23	0,00	0,00	0,00	0,22	2,00
	Flann Based	SURF	SURF	63	52	52	51	0,53	5,68	4,04	-0,27	-0,34	2,56	0,46	5,67	3,13	0,13	1,31
				63	48	48	47	0,46	3,69	3,35	-0,25	-0,64	2,27	0,38	3,64	2,47	0,13	1,31
				64	51	51	49	2,37	6,13	4,70	0,16	0,01	2,45	2,36	6,13	4,01	0,13	1,25
				63	48	48	46	2,71	6,42	4,98	0,22	0,10	2,33	2,70	6,42	4,40	0,13	1,25
			SIFT	33	20	20	20	0,43	0,67	2,19	-0,16	-0,61	0,93	0,41	0,28	1,99	0,29	1,30
				33	13	13	13	0,36	0,75	2,29	0,01	-0,61	1,56	0,36	0,44	1,69	0,29	1,23
				32	18	18	18	0,39	0,76	2,77	-0,10	-0,69	1,36	0,38	0,31	2,42	0,29	1,33
				32	14	14	14	0,50	0,91	2,28	-0,18	-0,79	0,65	0,47	0,45	2,18	0,29	1,29
		SIFT	SIFT	38	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,25	-nan
				38	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,25	-nan
				35	3	3	0	0,47	71,29	5,16	-0,40	70,80	4,57	0,23	8,37	2,40	0,27	1,00
				39	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,24	-nan

Rejection + Ratio

								Accuracy Test						Precision test				
	Matcher	Detetor	Extrator	Processed Images	Localization calculated	Localization not filtered	Localization calculated correctly	Standard Deviation x(cm)	Standard Deviation y(cm)	Standard Deviation theta(º)	Average Error x (cm)	Average Error y (cm)	Average Error theta (º)	Standard Deviation x(cm)	Standard Deviation y(cm)	Standard Deviation theta(º)	Average processing time(sec)	Average matches per image
Bag4	Brute Force	SURF	SURF	89	78	78	78	0,19	0,28	3,54	0,07	0,24	-2,78	0,18	0,14	2,19	0,10	1,09
			SIFT	34	30	28	28	0,27	0,27	4,70	0,14	0,25	-3,75	0,24	0,12	2,84	0,32	1,03
			BRISK	105	13	7	0	29,64	77,34	1,83	-29,64	77,34	1,50	0,24	0,27	1,05	0,09	1,00
		SIFT	SIFT	59	13	3	0	147,19	2,78	146,09	-138,70	2,33	-119,35	49,26	1,52	84,24	0,18	1,15
	Flann Based	SURF	SURF	66	59	57	57	0,19	0,29	3,75	0,10	0,26	-3,18	0,16	0,13	1,98	0,15	1,14
				66	57	57	57	0,18	0,29	3,78	0,11	0,26	-3,29	0,15	0,13	1,86	0,15	1,12
				67	58	56	56	0,17	0,27	3,18	0,05	0,22	-2,44	0,16	0,16	2,05	0,15	1,19
				66	58	56	56	0,24	0,30	4,32	0,14	0,27	-3,65	0,19	0,14	2,32	0,15	1,10
			SIFT	31	29	11	8	8,56	26,13	41,64	-1,82	13,38	14,77	8,36	22,45	38,94	0,39	1,48
				28	26	15	13	7,82	8,60	4,50	-1,90	2,45	-3,39	7,59	8,25	2,96	0,41	1,38
				29	26	7	6	6,55	20,32	35,43	-2,37	7,89	10,24	6,11	18,73	33,91	0,40	1,50
				29	28	11	11	0,22	0,24	3,86	0,16	0,24	-3,54	0,16	0,07	1,53	0,40	1,43
		SIFT	SIFT	53	12	5	0	69,44	0,15	0,59	-69,44	0,15	-0,52	0,32	0,04	0,28	0,21	1,00
				51	11	6	0	115,05	2,12	102,75	-104,13	1,32	-59,74	48,92	1,66	83,60	0,21	1,09
				49	13	9	0	156,48	3,02	157,72	-150,36	2,70	-139,20	43,30	1,36	74,14	0,22	1,00
				12	7	4	0	77,55	36,23	118,27	77,55	36,23	-118,25	0,32	0,27	2,14	0,26	1,57
Bag5	Brute Force	SURF	SURF	94	47	47	47	1,38	0,82	3,73	-0,84	0,40	-0,01	1,09	0,71	3,73	0,09	1,15
			SIFT	40	40	40	40	0,72	0,24	1,95	-0,63	0,00	-0,95	0,35	0,24	1,71	0,27	4,65
			BRISK	104	75	59	59	0,57	0,50	1,50	-0,54	0,49	-0,70	0,18	0,11	1,32	0,08	1,17
		SIFT	SIFT	42	33	23	22	1,20	0,63	4,12	-0,06	0,50	-2,02	1,19	0,38	3,60	0,25	1,61
	Flann Based	SURF	SURF	69	41	41	41	1,40	0,98	3,95	-0,90	0,57	-0,01	1,07	0,79	3,95	0,14	1,17
				69	39	39	39	1,32	0,87	3,48	-0,97	0,46	0,38	0,89	0,73	3,46	0,14	1,21
				69	34	34	34	1,35	0,71	3,19	-0,99	0,31	0,61	0,91	0,63	3,13	0,14	1,15
				67	37	37	37	1,34	0,79	3,54	-0,97	0,40	0,29	0,92	0,68	3,53	0,15	1,14
			SIFT	30	30	30	29	1,41	1,82	6,17	-0,92	0,12	-2,00	1,07	1,82	5,83	0,37	5,00
				30	30	30	28	1,54	1,50	7,00	-0,57	0,12	-2,64	1,44	1,50	6,48	0,37	5,07
				29	29	27	27	0,87	0,27	1,17	-0,83	-0,15	0,09	0,23	0,23	1,17	0,38	5,52
				29	29	27	25	1,87	2,49	9,34	-1,03	0,76	-3,50	1,56	2,38	8,66	0,38	5,10
		SIFT	SIFT	37	29	18	12	3,24	0,62	9,20	1,90	0,43	-7,10	2,62	0,45	5,85	0,29	1,59
				37	27	20	16	16,64	13,17	5,84	-4,64	-3,82	-3,21	15,98	12,60	4,87	0,29	1,48
				37	28	14	12	2,46	0,53	6,95	0,94	0,38	-3,94	2,27	0,38	5,72	0,29	1,39
				36	33	24	20	10,94	8,68	5,36	-2,78	-1,99	-3,18	10,58	8,44	4,32	0,30	1,76
Bag6	Brute Force	SURF	SURF	67	67	64	64	0,45	0,26	2,72	0,41	0,05	2,53	0,18	0,26	1,01	0,08	3,40
			SIFT	29	29	28	28	0,41	0,85	2,23	0,29	-0,60	1,49	0,30	0,61	1,66	0,23	3,66
			BRISK	74	40	29	26	3,15	3,22	3,64	1,25	-0,31	-0,89	2,89	3,21	3,53	0,07	1,32
		SIFT	SIFT	27	26	26	26	0,52	0,61	2,31	0,51	0,59	2,25	0,14	0,15	0,54	0,25	1,00
	Flann Based	SURF	SURF	44	44	44	44	0,45	0,29	2,71	0,40	0,00	2,54	0,21	0,29	0,94	0,14	3,43
				45	45	45	45	0,45	0,36	2,58	0,39	-0,03	2,35	0,23	0,36	1,06	0,14	3,47
				43	43	43	43	0,43	0,22	2,74	0,41	0,03	2,61	0,14	0,21	0,82	0,15	3,60
				43	43	41	41	0,48	0,30	2,72	0,43	0,04	2,58	0,23	0,30	0,89	0,14	3,51
			SIFT	21	21	21	21	1,68	1,22	1,86	-0,24	-0,45	1,32	1,66	1,13	1,31	0,33	4,86
				21	21	19	18	2,45	1,75	1,35	-0,42	-0,40	1,04	2,42	1,70	0,86	0,34	4,52
				21	21	19	18	2,43	1,69	2,15	-0,44	-0,22	1,80	2,39	1,67	1,19	0,33	4,86
				20	20	19	19	0,21	0,71	2,03	0,17	-0,64	1,65	0,12	0,32	1,19	0,35	4,50
		SIFT	SIFT	24	24	24	24	0,57	0,65	2,48	0,54	0,63	2,39	0,17	0,18	0,65	0,29	1,00
				24	24	20	20	0,62	0,71	2,65	0,58	0,67	2,55	0,20	0,21	0,73	0,29	1,08
				24	24	24	24	0,60	0,68	2,59	0,57	0,65	2,47	0,21	0,22	0,79	0,29	1,00
				24	23	21	21	0,61	0,71	2,65	0,58	0,67	2,54	0,20	0,21	0,76	0,29	1,04

Results

								Accuracy Test						Precision test				
	Matcher	Detetor	Extrator	Processed Images	Localization calculated	Localization not filtered	Localization calculated correctly	Standard Deviation x(cm)	Standard Deviation y(cm)	Standard Deviation theta(º)	Average Error x (cm)	Average Error y (cm)	Average Error theta (º)	Standard Deviation x(cm)	Standard Deviation y(cm)	Standard Deviation theta(º)	Average processing time(sec)	Average matches per image
Bag7	Brute Force	SURF	SURF	78	58	58	56	1,90	1,37	3,35	-1,75	-0,99	0,98	0,74	0,93	3,20	0,09	1,09
			SIFT	37	36	35	35	1,46	0,77	1,64	-1,39	-0,58	-0,26	0,45	0,50	1,62	0,22	1,89
			BRISK	92	22	21	21	1,81	0,93	2,51	-1,79	-0,85	2,30	0,20	0,38	1,02	0,07	1,00
		SIFT	SIFT	32	32	31	31	1,72	1,26	3,43	-1,62	-0,43	1,26	0,58	1,18	3,19	0,26	1,91
	Flann Based	SURF	SURF	54	44	44	44	1,84	1,01	2,50	-1,73	-0,88	0,82	0,63	0,50	2,36	0,14	1,20
				54	41	41	41	2,01	1,10	2,77	-1,93	-1,03	1,60	0,59	0,39	2,26	0,14	1,15
				55	41	41	41	1,96	1,05	2,59	-1,86	-0,91	1,30	0,60	0,54	2,24	0,14	1,24
				53	41	41	41	1,85	1,02	2,21	-1,78	-0,92	1,01	0,53	0,43	1,97	0,14	1,07
			SIFT	26	26	26	26	1,62	0,89	1,81	-1,54	-0,84	0,65	0,49	0,31	1,69	0,33	2,15
				26	24	24	22	2,92	2,74	3,57	-2,06	-0,93	-0,32	2,07	2,58	3,56	0,32	2,96
				26	25	25	23	3,24	1,35	1,50	-2,17	-0,40	0,49	2,40	1,29	1,42	0,32	2,52
				26	25	24	24	1,53	0,85	1,74	-1,47	-0,77	0,75	0,42	0,35	1,56	0,32	3,20
		SIFT	SIFT	29	29	25	25	1,62	1,07	2,86	-1,55	-0,39	1,04	0,47	0,99	2,67	0,29	2,21
				31	31	31	31	1,73	1,26	3,45	-1,63	-0,49	1,39	0,58	1,16	3,16	0,28	1,94
				31	30	28	27	1,74	1,58	4,20	-1,62	-0,38	1,13	0,64	1,54	4,05	0,28	2,10
				29	29	28	28	1,63	0,99	2,68	-1,57	-0,41	1,16	0,44	0,91	2,42	0,29	2,21
Bag8	Brute Force	SURF	SURF	59	37	28	19	1,53	3,16	18,01	0,74	-1,13	0,76	1,34	2,95	17,99	0,11	1,22
			SIFT	22	22	21	21	0,38	0,18	0,55	0,35	-0,08	-0,11	0,16	0,15	0,54	0,36	10,50
			BRISK	68	9	9	9	2,14	2,20	6,10	2,08	-1,69	5,77	0,52	1,41	2,00	0,09	1,00
		SIFT	SIFT	37	1	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,20	1,00
	Flann Based	SURF	SURF	45	33	31	23	0,98	1,68	7,08	0,54	-1,23	1,68	0,82	1,15	6,87	0,16	1,36
				46	30	23	22	1,07	1,33	5,06	0,54	-0,97	-0,28	0,93	0,92	5,05	0,15	1,67
				46	33	27	24	1,31	2,83	17,50	0,20	-0,35	-3,89	1,29	2,81	17,06	0,15	1,58
				44	33	16	11	5,91	6,00	45,12	2,04	0,35	-77,79	5,55	6,00	80,54	0,16	1,45
			SIFT	19	19	19	17	3,27	2,73	3,29	1,77	0,82	0,25	2,76	2,60	3,28	0,43	14,11
				19	19	19	17	3,31	2,31	2,94	1,47	0,77	1,20	2,96	2,18	2,69	0,43	13,79
				19	19	19	16	3,97	2,84	3,58	1,97	1,17	1,82	3,45	2,58	3,09	0,43	13,58
				19	19	18	16	3,53	2,20	4,71	1,43	0,68	1,08	3,23	2,10	4,58	0,43	13,63
		SIFT	SIFT	34	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,22	-nan
				32	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,24	-nan
				32	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,24	-nan
				33	2	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,23	1,00
Bag9	Brute Force	SURF	SURF	106	75	72	72	0,37	1,14	2,61	-0,18	-1,13	1,76	0,33	0,18	1,93	0,07	1,09
			SIFT	53	15	14	14	0,58	0,71	3,52	0,27	-0,43	2,38	0,52	0,57	2,60	0,17	1,40
			BRISK	116	67	36	34	36,65	27,11	21,49	8,02	5,16	0,91	35,76	26,62	21,47	0,06	1,13
		SIFT	SIFT	42	1	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,22	1,00
	Flann Based	SURF	SURF	63	49	47	47	0,38	1,22	2,45	-0,19	-1,20	1,43	0,32	0,23	1,99	0,14	1,18
				63	55	53	53	0,41	1,15	2,81	-0,21	-1,13	1,95	0,36	0,18	2,02	0,14	1,18
				64	54	52	52	0,14	1,17	2,44	-0,11	-1,15	1,68	0,08	0,24	1,77	0,13	1,19
				63	56	53	53	0,30	1,16	2,56	-0,18	-1,14	1,96	0,24	0,21	1,64	0,13	1,23
			SIFT	33	17	17	17	0,54	0,75	3,17	-0,13	-0,65	0,87	0,53	0,38	3,05	0,29	1,06
				33	17	17	17	0,45	0,94	2,41	-0,13	-0,82	0,72	0,42	0,47	2,30	0,29	1,29
				33	16	16	16	0,27	0,85	2,02	-0,05	-0,74	1,20	0,26	0,41	1,62	0,29	1,31
				32	18	17	17	0,38	0,82	2,89	-0,02	-0,69	1,16	0,37	0,45	2,65	0,30	1,44
		SIFT	SIFT	38	2	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,24	1,00
				37	1	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,25	1,00
				37	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,25	-nan
				38	1	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,25	2,00

Rejection + Ransac

	Matcher	Detetor	Extrator	Processed Images	Localization calculated	Localization not filtered	Localization calculated correctly	Accuracy Test: Standard Deviation x(cm)	Accuracy Test: Standard Deviation y(cm)	Accuracy Test: Standard Deviation theta(º)	Accuracy Test: Average Error x (cm)	Accuracy Test: Average Error y (cm)	Accuracy Test: Average Error theta (º)	Precision test: Standard Deviation x(cm)	Precision test: Standard Deviation y(cm)	Precision test: Standard Deviation theta(º)	Average processing time(sec)	Average matches per image
Bag4	Brute Force	SURF	SURF	73	1	1	1	0,13	0,24	3,64	0,13	0,24	-3,64	0,00	0,00	0,00	0,13	1,00
			SIFT	32	3	3	3	0,27	0,30	5,22	0,25	0,29	-5,04	0,11	0,06	1,37	0,35	1,00
			BRISK	80	1	1	0	29,55	77,24	1,10	-29,55	77,24	1,10	0,00	0,00	0,00	0,12	1,00
		SIFT	SIFT	53	1	1	0	171,90	31,37	91,07	-171,90	31,37	91,07	0,00	0,00	0,00	0,20	1,00
	Flann Based	SURF	SURF	57	3	3	3	0,24	0,28	4,65	0,20	0,27	-4,40	0,12	0,05	1,50	0,18	1,00
				58	1	1	1	0,17	0,11	0,08	-0,17	0,11	-0,08	0,00	0,00	0,00	0,18	1,00
				56	2	2	2	0,11	0,22	3,20	0,10	0,22	-3,12	0,06	0,03	0,71	0,18	1,00
				57	3	3	3	0,16	0,41	3,24	0,16	0,37	-3,15	0,02	0,18	0,75	0,18	1,00
			SIFT	27	2	2	0	6,06	103,63	179,45	-6,06	103,63	0,39	0,10	0,07	179,45	0,42	1,00
				29	4	4	1	5,24	89,73	155,46	-4,48	77,78	-135,77	2,71	44,74	75,72	0,43	1,00
				26	5	5	0	6,00	103,55	178,92	-6,00	103,55	-178,92	0,02	0,05	0,50	0,43	1,40
				27	2	2	0	6,10	103,60	179,54	-6,10	103,60	-179,54	0,06	0,04	0,34	0,42	1,00
		SIFT	SIFT	47	2	2	0	131,99	2,49	126,27	-121,23	1,85	-89,39	52,19	1,66	89,18	0,23	1,00
				47	2	2	0	132,22	2,45	126,39	-121,63	1,80	-89,72	51,84	1,67	89,01	0,23	1,00
				48	1	1	0	5,78	58,13	1,27	-5,78	58,13	1,27	0,00	0,00	0,00	0,22	1,00
				47	5	5	0	151,64	15,56	138,74	-146,21	-4,68	-112,74	40,21	14,84	80,86	0,23	1,00
Bag5	Brute Force	SURF	SURF	77	23	23	23	1,00	1,10	3,16	-0,75	0,84	-0,83	0,67	0,70	3,05	0,12	1,09
			SIFT	35	27	27	27	0,98	0,40	3,01	-0,71	-0,18	-0,51	0,69	0,37	2,96	0,30	1,81
			BRISK	82	5	5	4	1,19	0,69	5,72	0,03	0,18	-2,98	1,19	0,67	4,88	0,11	1,00
		SIFT	SIFT	38	16	16	14	24,94	20,47	3,79	-6,13	-4,97	-1,68	24,18	19,86	3,39	0,28	1,56
	Flann Based	SURF	SURF	59	15	15	15	1,23	1,02	3,33	-0,94	0,75	-0,25	0,79	0,70	3,32	0,17	1,13
				58	15	15	15	2,00	0,82	5,25	-1,69	0,31	3,01	1,07	0,76	4,30	0,17	1,00
				59	15	15	15	1,24	1,06	3,24	-0,96	0,78	-0,30	0,78	0,72	3,23	0,17	1,07
				58	11	11	11	1,05	1,12	3,35	-0,82	0,78	-0,34	0,66	0,79	3,33	0,17	1,09
			SIFT	28	18	18	18	0,83	0,43	2,62	-0,65	-0,02	-0,76	0,51	0,43	2,51	0,40	2,39
				28	19	19	19	1,36	0,52	3,83	-1,06	0,01	0,77	0,86	0,52	3,75	0,40	1,89
				27	18	18	18	0,88	0,63	2,58	-0,66	-0,18	-0,67	0,59	0,60	2,49	0,41	1,72
				28	18	18	17	1,19	0,43	3,81	-0,77	-0,15	-0,36	0,90	0,40	3,79	0,40	2,72
		SIFT	SIFT	34	11	11	9	33,04	26,86	3,35	-13,01	-10,53	-1,88	30,37	24,72	2,78	0,32	1,45
				35	14	14	11	37,02	29,90	23,84	-16,29	-13,13	4,53	33,25	26,86	23,41	0,31	1,64
				35	16	16	15	13,63	11,18	1,73	-3,79	-2,44	-0,84	13,10	10,90	1,51	0,32	1,25
				35	13	13	10	27,32	22,67	6,16	-6,71	-6,17	-3,60	26,48	21,81	5,00	0,31	1,31
Bag6	Brute Force	SURF	SURF	55	54	54	53	8,70	3,83	15,82	-0,77	0,52	0,32	8,66	3,79	15,81	0,11	3,15
			SIFT	26	23	23	22	13,44	5,64	23,35	-2,68	0,37	-3,52	13,17	5,62	23,09	0,26	2,74
			BRISK	60	16	16	16	0,34	1,57	1,58	0,33	-1,55	-1,26	0,07	0,27	0,96	0,10	1,19
		SIFT	SIFT	27	14	14	13	17,09	7,27	30,06	-4,07	2,45	-5,83	16,60	6,85	29,49	0,26	1,07
	Flann Based	SURF	SURF	38	37	37	37	0,44	0,45	2,67	0,36	-0,02	2,40	0,24	0,45	1,16	0,17	2,76
				38	38	38	37	1,11	1,02	20,75	0,18	0,10	5,61	1,10	1,02	19,98	0,17	3,00
				38	38	38	37	1,53	0,45	24,42	0,11	-0,04	6,39	1,53	0,45	23,57	0,17	3,08
				39	39	39	39	0,48	0,57	2,75	0,41	-0,04	2,42	0,25	0,57	1,30	0,16	2,97
			SIFT	20	18	18	18	0,36	1,18	2,59	0,23	-0,88	1,51	0,28	0,78	2,10	0,36	2,50
				20	15	15	15	0,46	1,35	2,98	0,25	-0,89	1,39	0,38	1,01	2,63	0,36	3,13
				20	17	17	17	0,45	1,02	2,12	0,20	-0,66	1,32	0,40	0,78	1,65	0,36	2,65
				20	15	15	14	16,66	6,83	28,54	-4,18	1,21	-5,43	16,12	6,72	28,02	0,36	2,67
		SIFT	SIFT	24	15	15	15	0,59	0,70	2,54	0,55	0,57	2,36	0,23	0,40	0,93	0,29	1,07
				23	15	15	15	0,66	0,74	2,78	0,61	0,70	2,64	0,23	0,24	0,86	0,30	1,00
				23	12	12	12	0,58	0,67	2,53	0,55	0,64	2,44	0,18	0,19	0,69	0,30	1,00
				22	14	14	13	17,14	7,15	30,17	-4,04	2,52	-5,71	16,66	6,70	29,62	0,31	1,14

								Accuracy Test						Precision test				
	Matcher	Detetor	Extrator	Processed Images	Localization calculated	Localization not filtered	Localization calculated correctly	Standard Deviation x(cm)	Standard Deviation y(cm)	Standard Deviation theta(º)	Average Error x (cm)	Average Error y (cm)	Average Error theta (º)	Standard Deviation x(cm)	Standard Deviation y(cm)	Standard Deviation theta(º)	Average processing time(sec)	Average matches per image
Bag7	Brute Force	SURF	SURF	53	52	52	52	0,48	0,46	2,73	0,41	-0,03	2,46	0,25	0,45	1,17	0,11	2,98
			SIFT	27	24	24	23	13,06	5,59	23,13	-2,48	0,21	-3,68	12,82	5,59	22,83	0,25	3,58
			BRISK	57	18	18	17	5,52	5,73	4,85	1,51	0,15	-0,35	5,31	5,72	4,84	0,10	1,17
		SIFT	SIFT	26	16	16	15	15,77	6,93	29,42	-3,36	2,19	-5,03	15,41	6,57	28,99	0,26	1,12
	Flann Based	SURF	SURF	38	38	38	38	0,43	0,37	2,60	0,37	-0,06	2,39	0,21	0,37	1,02	0,17	3,34
				38	38	38	38	0,35	0,45	2,54	0,31	-0,17	2,31	0,16	0,41	1,07	0,17	2,84
				38	38	38	38	0,44	0,52	2,69	0,36	-0,12	2,39	0,24	0,51	1,24	0,17	3,13
				38	38	38	38	0,48	0,63	2,67	0,39	-0,14	2,31	0,29	0,61	1,35	0,17	3,03
			SIFT	20	15	15	15	0,27	0,88	3,51	0,11	-0,35	2,82	0,24	0,80	2,09	0,35	2,87
				19	14	14	14	0,57	1,30	3,25	0,28	-0,81	1,73	0,50	1,01	2,75	0,36	2,57
				20	17	17	16	26,30	18,49	3,31	-6,18	3,77	1,01	25,56	18,10	3,16	0,36	2,82
				21	16	16	15	16,06	6,80	28,01	-3,74	0,96	-5,59	15,62	6,73	27,45	0,36	3,00
		SIFT	SIFT	24	13	13	11	35,58	30,71	33,37	-13,52	10,25	10,88	32,92	28,95	31,55	0,30	1,08
				23	14	14	13	15,23	3,14	2,56	-3,53	1,43	2,32	14,81	2,79	1,08	0,30	1,07
				23	11	11	11	0,61	0,69	2,60	0,56	0,64	2,44	0,24	0,26	0,91	0,30	1,00
				23	17	17	17	0,66	0,76	2,83	0,63	0,72	2,73	0,20	0,21	0,76	0,30	1,00
Bag8	Brute Force	SURF	SURF	46	4	4	3	11,74	11,71	89,86	6,59	4,94	-44,38	9,71	10,61	78,14	0,15	1,50
			SIFT	20	20	20	19	13,94	1,82	34,76	3,66	-0,21	8,33	13,45	1,81	33,74	0,39	7,85
			BRISK	53	2	2	2	1,62	0,60	3,41	1,61	0,60	3,40	0,10	0,04	0,23	0,13	1,00
		SIFT	SIFT	33	1	1	0	62,42	7,79	154,99	62,42	-7,79	154,99	0,00	0,00	0,00	0,23	1,00
	Flann Based	SURF	SURF	38	7	7	5	2,16	5,23	30,88	-0,42	0,96	-10,80	2,12	5,14	28,93	0,19	1,43
				39	3	3	2	0,93	1,53	7,96	-0,14	-0,73	-0,96	0,92	1,35	7,90	0,19	1,00
				38	4	4	3	11,69	11,73	89,68	6,42	4,67	-44,64	9,77	10,76	77,77	0,19	1,75
				39	6	6	3	1,12	2,23	9,49	0,84	-1,85	5,63	0,73	1,24	7,65	0,18	1,17
			SIFT	18	18	18	18	0,92	0,44	1,55	0,88	0,40	1,43	0,26	0,16	0,61	0,46	8,33
				18	18	18	18	0,82	0,38	1,38	0,76	0,35	1,12	0,31	0,15	0,81	0,46	8,50
				18	18	18	18	0,79	0,37	1,26	0,73	0,29	1,02	0,29	0,22	0,74	0,46	8,44
				18	18	18	16	14,79	3,02	36,69	4,27	-0,88	7,73	14,16	2,89	35,87	0,45	9,56
		SIFT	SIFT	30	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,25	-nan
				29	2	2	0	67,16	48,73	90,71	6,78	-34,05	-90,66	66,82	34,86	3,12	0,26	1,00
				30	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,25	-nan
				28	2	2	0	68,21	5,56	125,46	67,98	-3,26	34,03	5,54	4,50	120,76	0,27	1,00
Bag9	Brute Force	SURF	SURF	82	53	53	53	0,36	1,12	2,86	-0,18	-1,11	1,92	0,30	0,14	2,12	0,10	1,13
			SIFT	46	10	10	9	24,28	11,26	36,70	7,78	3,08	-10,15	23,00	10,82	35,26	0,20	1,30
			BRISK	87	30	30	29	2,03	11,48	10,53	-0,27	0,83	-2,30	2,02	11,45	10,28	0,09	1,03
		SIFT	SIFT	42	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,22	-nan
	Flann Based	SURF	SURF	53	42	42	42	0,18	1,15	2,72	-0,13	-1,13	2,05	0,13	0,22	1,79	0,16	1,24
				54	34	34	34	0,40	1,15	2,88	-0,21	-1,13	2,03	0,34	0,19	2,04	0,16	1,32
				54	42	42	42	0,17	1,12	2,44	-0,14	-1,11	1,50	0,10	0,13	1,92	0,16	1,19
				55	36	36	36	0,30	1,16	2,32	-0,18	-1,14	1,40	0,25	0,16	1,85	0,16	1,19
			SIFT	31	3	3	3	0,64	0,82	3,33	-0,41	-0,80	-0,67	0,50	0,20	3,27	0,31	1,33
				30	7	7	7	0,31	0,93	2,21	-0,16	-0,90	1,24	0,27	0,24	1,83	0,32	1,43
				30	8	8	8	0,13	0,94	3,05	0,01	-0,84	1,03	0,13	0,43	2,87	0,32	1,12
				29	6	6	5	31,74	14,90	49,40	12,72	5,28	-20,34	29,08	13,94	45,02	0,33	1,17
		SIFT	SIFT	38	1	1	0	0,42	65,11	3,81	-0,42	65,11	3,81	0,00	0,00	0,00	0,25	1,00
				37	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,25	-nan
				38	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,24	-nan
				36	1	1	0	77,34	36,36	117,18	77,34	36,36	-117,18	0,00	0,00	0,00	0,26	1,00

Results

Ratio + Rejection + RANSAC

	Matcher	Detetor	Extrator	Processed Images	Localization calculated	Localization not filtered	Localization calculated correctly	Accuracy Test: Standard Deviation x(cm)	Accuracy Test: Standard Deviation y(cm)	Accuracy Test: Standard Deviation theta(º)	Accuracy Test: Average Error x (cm)	Accuracy Test: Average Error y (cm)	Accuracy Test: Average Error theta (º)	Precision test: Standard Deviation x(cm)	Precision test: Standard Deviation y(cm)	Precision test: Standard Deviation theta(º)	Average processing time(sec)	Average matches per image
Bag4	Brute Force	SURF	SURF	72	5	5	5	0,15	0,21	2,79	0,02	0,19	-2,24	0,14	0,07	1,66	0,14	1,00
			SIFT	32	2	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,35	1,00
			BRISK	81	1	1	1	3,20	0,26	6,92	-3,20	0,26	6,92	0,00	0,00	0,00	0,12	1,00
		SIFT	SIFT	54	2	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,20	1,00
	Flann Based	SURF	SURF	58	1	1	1	0,04	0,17	1,77	-0,04	0,17	-1,77	0,00	0,00	0,00	0,18	1,00
				57	1	1	1	0,38	0,35	6,52	0,38	0,35	-6,52	0,00	0,00	0,00	0,18	1,00
				58	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,18	-nan
				58	3	2	2	0,28	0,30	5,21	0,25	0,29	-4,98	0,13	0,05	1,54	0,18	1,00
			SIFT	27	10	2	0	6,09	103,63	179,49	-6,09	103,63	-179,66	0,07	0,07	0,51	0,42	1,20
				27	6	3	1	4,91	84,55	146,03	-3,97	69,12	-120,52	2,91	48,70	82,47	0,42	1,33
				27	8	3	1	4,81	84,44	145,15	-3,93	69,00	-119,16	2,77	48,66	82,88	0,43	1,00
				27	4	2	0	6,04	103,58	179,28	-6,04	103,58	-179,28	0,05	0,02	0,19	0,42	1,00
		SIFT	SIFT	47	4	2	0	69,08	0,20	0,15	-69,08	0,20	-0,12	0,04	0,01	0,09	0,23	1,00
				46	3	1	0	69,79	0,13	0,71	-69,79	0,13	-0,71	0,00	0,00	0,00	0,23	1,00
				46	6	1	0	69,76	0,15	0,57	-69,76	0,15	-0,57	0,00	0,00	0,00	0,23	1,17
				46	6	2	0	132,00	2,39	126,58	-121,17	1,75	-89,89	52,38	1,63	89,11	0,24	1,00
Bag5	Brute Force	SURF	SURF	95	41	41	41	1,35	0,71	3,16	-1,11	0,34	0,85	0,78	0,62	3,05	0,09	1,15
			SIFT	36	30	30	30	0,71	0,47	2,99	-0,49	-0,01	-1,73	0,51	0,47	2,43	0,30	2,53
			BRISK	83	12	12	11	0,82	0,72	3,77	-0,30	0,46	-1,91	0,76	0,56	3,25	0,11	1,00
		SIFT	SIFT	39	18	16	14	1,85	0,38	5,02	0,23	0,26	-1,59	1,84	0,28	4,76	0,28	1,28
	Flann Based	SURF	SURF	58	14	14	14	1,50	0,89	4,22	-1,16	0,15	1,51	0,94	0,88	3,94	0,17	1,07
				59	25	25	25	1,32	0,85	3,85	-0,85	0,56	-0,01	1,02	0,63	3,85	0,17	1,12
				59	18	18	18	1,71	0,72	4,20	-1,43	0,34	2,05	0,93	0,64	3,67	0,17	1,06
				57	16	16	16	1,43	0,89	3,77	-1,12	0,50	0,87	0,89	0,74	3,67	0,18	1,12
			SIFT	28	19	19	19	1,04	0,46	2,08	-0,88	-0,09	0,02	0,55	0,46	2,08	0,40	2,42
				28	19	19	19	0,94	0,58	3,19	-0,71	-0,01	-0,83	0,61	0,58	3,08	0,40	2,47
				28	18	18	17	1,24	0,54	3,95	-0,82	-0,12	-0,19	0,94	0,54	3,94	0,40	2,06
				28	22	21	21	1,12	0,28	2,92	-0,86	-0,07	-0,21	0,71	0,26	2,91	0,40	2,18
		SIFT	SIFT	34	10	8	6	2,96	0,56	7,89	0,85	0,39	-4,25	2,84	0,40	6,65	0,32	1,30
				34	12	12	9	3,09	0,46	8,19	1,22	0,35	-4,31	2,84	0,32	6,97	0,32	1,08
				35	15	15	14	1,25	0,44	3,78	0,03	0,19	-1,44	1,25	0,40	3,50	0,31	1,33
				35	17	17	15	1,64	0,50	4,92	0,29	0,26	-2,52	1,62	0,43	4,23	0,31	1,47
Bag6	Brute Force	SURF	SURF	53	53	50	50	0,41	0,42	2,55	0,35	-0,06	2,27	0,20	0,41	1,16	0,11	2,92
			SIFT	26	25	24	24	0,23	1,02	2,04	0,18	-0,84	1,34	0,15	0,59	1,53	0,26	2,64
			BRISK	59	17	15	15	0,36	1,44	4,13	0,24	-1,17	0,33	0,27	0,84	4,12	0,10	1,24
		SIFT	SIFT	25	13	13	13	0,53	0,61	2,33	0,51	0,60	2,28	0,13	0,14	0,51	0,27	1,00
	Flann Based	SURF	SURF	38	38	38	38	0,66	0,67	3,17	0,44	-0,02	2,60	0,49	0,67	1,82	0,17	3,00
				39	39	39	39	0,44	0,52	2,71	0,38	-0,08	2,44	0,23	0,52	1,19	0,16	3,05
				39	39	39	39	0,36	0,44	2,59	0,32	-0,14	2,26	0,18	0,41	1,27	0,16	3,18
				38	37	37	37	0,42	0,46	2,63	0,35	-0,13	2,38	0,24	0,44	1,12	0,17	3,27
			SIFT	20	12	12	12	0,23	1,00	1,63	0,19	-0,88	1,20	0,14	0,47	1,11	0,36	3,25
				20	15	15	15	0,47	1,23	2,59	0,26	-0,85	1,22	0,38	0,89	2,29	0,35	2,73
				20	15	15	15	0,54	0,89	2,70	0,27	-0,40	1,92	0,47	0,79	1,90	0,36	2,13
				19	10	9	9	0,33	0,97	1,67	0,24	-0,77	1,29	0,23	0,60	1,05	0,37	3,70
		SIFT	SIFT	24	7	7	7	0,57	0,65	2,46	0,52	0,61	2,32	0,21	0,23	0,82	0,29	1,00
				23	12	12	12	0,63	0,71	2,70	0,60	0,68	2,58	0,21	0,22	0,80	0,30	1,00
				23	13	13	13	0,65	0,76	2,75	0,60	0,61	2,53	0,26	0,44	1,06	0,30	1,08
				23	18	15	15	0,58	0,68	2,49	0,53	0,55	2,31	0,23	0,40	0,94	0,30	1,22

								Accuracy Test						Precision test				
	Matcher	Detetor	Extrator	Processed Images	Localization calculated	Localization not filtered	Localization calculated correctly	Standard Deviation x(cm)	Standard Deviation y(cm)	Standard Deviation theta(º)	Average Error x (cm)	Average Error y (cm)	Average Error theta (º)	Standard Deviation x(cm)	Standard Deviation y(cm)	Standard Deviation theta(º)	Average processing time(sec)	Average matches per image
Bag7	Brute Force	SURF	SURF	65	4	4	4	2,20	1,73	4,98	-1,94	-0,46	1,25	1,03	1,66	4,82	0,11	1,25
			SIFT	33	12	11	11	1,71	0,88	2,34	-1,61	-0,52	0,34	0,58	0,71	2,31	0,25	1,25
			BRISK	73	1	1	1	1,75	0,74	2,02	-1,75	-0,74	2,02	0,00	0,00	0,00	0,09	1,00
		SIFT	SIFT	30	26	26	26	1,84	1,41	3,85	-1,73	-0,65	1,88	0,60	1,25	3,36	0,28	1,88
	Flann Based	SURF	SURF	47	13	13	13	1,84	1,38	3,16	-1,72	-0,63	0,45	0,65	1,22	3,12	0,17	1,00
				47	7	7	7	1,66	1,02	2,90	-1,47	-0,14	-0,43	0,76	1,02	2,87	0,16	1,14
				47	12	12	12	2,37	1,30	4,18	-2,22	-1,16	2,69	0,82	0,57	3,20	0,16	1,33
				47	12	12	12	1,90	1,42	3,46	-1,75	-0,61	0,51	0,75	1,28	3,42	0,16	1,08
			SIFT	24	13	13	13	1,83	0,96	2,43	-1,74	-0,91	1,34	0,58	0,30	2,03	0,35	1,46
				24	14	14	14	1,58	0,93	2,77	-1,45	-0,74	0,55	0,64	0,56	2,71	0,35	1,29
				24	11	11	11	1,77	0,98	2,06	-1,69	-0,95	1,72	0,52	0,26	1,13	0,35	1,36
				24	14	13	13	1,79	0,96	2,36	-1,73	-0,77	1,03	0,49	0,57	2,12	0,35	1,36
		SIFT	SIFT	26	26	26	26	1,79	1,21	3,36	-1,71	-0,63	1,82	0,53	1,04	2,83	0,33	1,81
				27	27	27	26	1,86	1,37	3,82	-1,75	-0,70	1,97	0,64	1,18	3,27	0,31	1,93
				27	26	26	25	1,92	1,43	4,01	-1,81	-0,81	2,32	0,64	1,18	3,27	0,31	1,88
				27	24	23	22	1,85	1,42	3,95	-1,72	-0,61	1,79	0,69	1,28	3,52	0,31	1,75
Bag8	Brute Force	SURF	SURF	47	4	4	2	1,16	2,11	7,83	1,12	-1,96	5,87	0,30	0,79	5,19	0,15	1,00
			SIFT	1	1	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,26	4,00
			BRISK	54	3	3	3	1,86	1,66	5,16	1,84	-0,52	4,77	0,33	1,57	1,95	0,13	1,00
		SIFT	SIFT	36	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,22	-nan
	Flann Based	SURF	SURF	39	5	5	3	0,94	1,94	7,94	0,54	-1,57	3,23	0,77	1,13	7,25	0,19	1,20
				39	3	3	3	0,52	0,78	3,93	-0,07	-0,57	-3,32	0,52	0,54	2,10	0,19	1,33
				40	4	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,18	1,00
				38	2	2	0	1,19	2,79	11,35	1,19	-2,79	11,32	0,04	0,12	0,84	0,19	1,00
			SIFT	18	18	18	18	0,81	0,39	1,29	0,75	0,35	1,02	0,33	0,18	0,79	0,46	8,22
				18	18	18	18	0,87	0,39	1,35	0,81	0,35	1,15	0,32	0,18	0,71	0,46	8,06
				17	17	17	17	0,77	0,38	1,23	0,71	0,35	0,97	0,29	0,17	0,74	0,46	8,41
				18	18	18	18	0,66	0,34	0,91	0,63	0,28	0,78	0,18	0,18	0,48	0,46	8,89
		SIFT	SIFT	30	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,26	-nan
				30	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,25	-nan
				30	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,26	-nan
				29	1	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,26	1,00
Bag9	Brute Force	SURF	SURF	83	50	50	50	0,14	1,18	2,09	-0,13	-1,16	1,59	0,06	0,21	1,36	0,10	1,12
			SIFT	46	9	8	8	0,35	0,70	2,04	-0,04	-0,57	1,08	0,35	0,41	1,73	0,20	1,11
			BRISK	86	32	23	22	4,08	24,08	18,95	0,20	3,75	-0,39	4,07	23,78	18,94	0,09	1,03
		SIFT	SIFT	43	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,21	-nan
	Flann Based	SURF	SURF	54	37	37	37	0,31	1,16	2,39	-0,15	-1,13	1,62	0,27	0,23	1,76	0,16	1,16
				54	41	41	41	0,21	1,16	2,76	-0,13	-1,13	1,81	0,16	0,22	2,08	0,16	1,20
				54	42	42	42	0,18	1,14	2,53	-0,15	-1,13	1,69	0,10	0,18	1,88	0,16	1,14
				55	43	43	43	0,15	1,13	2,36	-0,13	-1,12	1,78	0,07	0,21	1,54	0,16	1,12
			SIFT	30	7	7	7	0,10	0,78	3,21	-0,01	-0,69	1,59	0,10	0,37	2,80	0,32	1,14
				29	8	8	8	0,35	0,68	2,75	0,10	-0,54	1,42	0,33	0,40	2,36	0,33	1,12
				30	11	11	11	0,24	0,88	2,67	-0,13	-0,82	1,62	0,20	0,33	2,12	0,32	1,36
				30	7	7	7	0,62	1,12	2,45	-0,48	-1,02	1,09	0,39	0,47	2,19	0,32	1,29
		SIFT	SIFT	38	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,25	-nan
				38	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,25	-nan
				36	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,26	-nan
				38	0	0	0	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	-nan	0,24	-nan

Bibliography

[1] F. Bergholm, P. Trahanias and S. Orphanoudakis. Analysis of current approaches in automated vision-based navigation. pages 18 – 60, 1997.

[2] G. N. Desouza and A. C. Kak. Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, Feb 2002. doi:10.1109/34.982903.

[3] Christian Schlaile, Oliver Meister, Natalie Frietsch, Christoph Keßler, Jan Wendel, and Gert F. Trommer. Using natural features for vision based navigation of an indoor-vtol mav. *Aerospace Science and Technology*, 13(7):349 – 357, 2009. URL: http://www.sciencedirect.com/science/article/pii/S1270963809000522, doi:http://dx.doi.org/10.1016/j.ast.2009.09.001.

[4] K. Nagahama, K. Hashimoto, T. Noritsugu, and M. Takaiawa. Visual servoing based on object motion estimation. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, volume 1, pages 245–250 vol.1, 2000. doi:10.1109/IROS.2000.894612.

[5] X. Gao, L. Mo, D. You, and Z. Li. Tight butt joint weld detection based on optical flow and particle filtering of magneto-optical imaging. *Mechanical Systems and Signal Processing*, 96:16–30, 2017. cited By 0. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85019134542&doi=10.1016%2fj.ymssp.2017.04.001&partnerID=40&md5=b4d7857964956a5389d75045d75dd6a8, doi:10.1016/j.ymssp.2017.04.001.

[6] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 2, pages 2051–2058 vol.2, 2001. doi:10.1109/ROBOT.2001.932909.

[7] D. Santosh Kumar. Vision-based robot navigation using an online visual experience, 2007.

[8] B. Crespi, C. Furlanello, and L. Stringa. A memory-based approach to navigation. *Biological Cybernetics*, 69(5):385–393, 1993. URL: http://dx.doi.org/10.1007/BF00199438, doi:10.1007/BF00199438.

[9] R. C. Nelson. Visual homing using an associative memory. *Biological Cybernetics*, 65(4):281–291, 1991. URL: http://dx.doi.org/10.1007/BF00206225, doi:10.1007/BF00206225.

[10] Zhichao Chen and S. T. Birchfield. Qualitative vision-based mobile robot navigation. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2686–2692, May 2006. doi:10.1109/ROBOT.2006.1642107.

[11] Y. Mezouar, A. Remazeilles, P. Gros, and F. Chaumette. Images interpolation for image-based control under large displacement. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 4, pages 3787–3794 vol.4, 2002. doi:10.1109/ROBOT.2002.1014306.

[12] P. De Cristóforis, M. Nitsche, T. Krajník, T. Pire, and M. Mejail. Hybrid vision-based navigation for mobile robots in mixed indoor/outdoor environments. *Pattern Recognition Letters*, 53:118–128, 2015. cited By 0. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84922541097&doi=10.1016%2fj.patrec.2014.10.010&partnerID=40&md5=b4e290d79869080ddbc3269310869e7c, doi:10.1016/j.patrec.2014.10.010.

[13] Last accessed in 2017/5/20. URL: http://www.dictionary.com/browse/contour.

[14] Bill Green. Canny edge detector, 2002. Last accessed in 2017/6/10. URL: http://masters.donntu.org/2010/fknt/chudovskaja/library/article5.htm.

[15] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, February 1962. doi:10.1109/TIT.1962.1057692.

[16] Christoph Dalitz, Christian Brandt, Steffen Goebbels, and David Kolanus. Fourier descriptors for broken shapes. *EURASIP Journal on Advances in Signal Processing*, 2013(1):161, 2013. URL: http://dx.doi.org/10.1186/1687-6180-2013-161, doi:10.1186/1687-6180-2013-161.

[17] OpenCV. Shape matching, May 2017. Last accessed in2017/5/12. URL: http://docs.opencv.org/2.4.13.2/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html#matchshapes.

[18] Last accessed in 2017/5/20. URL: http://dictionary.cambridge.org/pt/dicionario/ingles/template.

[19] OpenCV. Template matching, May 2017. Last accessed in 2017/5/12. URL: http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html.

[20] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, page 147–151, 1988.

[21] Wikipedia. Difference of Gaussians — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Difference%20of%20Gaussians&oldid=747011119, 2017. [Online; accessed 10-June-2017].

[22] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555, Nov 2011. doi:10.1109/ICCV.2011.6126542.

[23] Edward Rosten and Tom Drummond. *Machine Learning for High-Speed Corner Detection*, pages 430–443. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. URL: http://dx.doi.org/10.1007/11744023_34, doi:10.1007/11744023_34.

[24] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary robust independent elementary features. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 778–792, Berlin, Heidelberg, 2010. Springer-Verlag. URL: http://dl.acm.org/citation.cfm?id=1888089.1888148.

[25] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT] or SURF, booktitle = Proceedings of the 2011 International Conference on Computer Vision, series = ICCV '11, year = 2011, isbn = 978-1-4577-1101-5, pages = 2564–2571, numpages = 8, url = http://dx.doi.org/10.1109/ICCV.2011.6126544, doi = 10.1109/ICCV.2011.6126544, acmid = 2356268, publisher = IEEE Computer Society, address = Washington, DC, USA,.

[26] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999. doi:10.1109/ICCV.1999.790410.

[27] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008. URL: http://dx.doi.org/10.1016/j.cviu.2007.09.014, doi:10.1016/j.cviu.2007.09.014.

[28] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 36.1–36.10. BMVA Press, 2002. doi:10.5244/C.16.36.

[29] A. Alahi, R. Ortiz, and P. Vandergheynst. FREAK: Fast retina keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517, June 2012. doi:10.1109/CVPR.2012.6247715.

[30] Arduino Playground. Reading rotary encoders, January 2010. Last accessed in 2016/12/12. URL: http://playground.arduino.cc/Main/RotaryEncoders.

[31] Jean-Yves Bouguet. Matlab camera calibration toolbox, October 2015. Last accessed in 2016/12/12. URL: http://www.vision.caltech.edu/bouguetj/calib_doc/.

[32] Hugin. Hugin - panorama photo stitcher, September 2016. Last accessed in2017/5/12. URL: http://hugin.sourceforge.net/.

[33] OpenCV. Features in opencv, May 2017. Last accessed in 2017/6/2. URL: http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_table_of_contents_feature2d/py_table_of_contents_feature2d.html.

References

[1] F. Bergholm, P. Trahanias and S. Orphanoudakis. Analysis of current approaches in automated vision-based navigation. pages 18 – 60, 1997.

[2] G. N. Desouza and A. C. Kak. Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, Feb 2002. doi:10.1109/34.982903.

[3] Christian Schlaile, Oliver Meister, Natalie Frietsch, Christoph Keßler, Jan Wendel, and Gert F. Trommer. Using natural features for vision based navigation of an indoor-vtol mav. *Aerospace Science and Technology*, 13(7):349 – 357, 2009. URL: http://www.sciencedirect.com/science/article/pii/S1270963809000522, doi:http://dx.doi.org/10.1016/j.ast.2009.09.001.

[4] K. Nagahama, K. Hashimoto, T. Noritsugu, and M. Takaiawa. Visual servoing based on object motion estimation. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, volume 1, pages 245–250 vol.1, 2000. doi:10.1109/IROS.2000.894612.

[5] X. Gao, L. Mo, D. You, and Z. Li. Tight butt joint weld detection based on optical flow and particle filtering of magneto-optical imaging. *Mechanical Systems and Signal Processing*, 96:16–30, 2017. cited By 0. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85019134542&doi=10.1016%2fj.ymssp.2017.04.001&partnerID=40&md5=b4d7857964956a5389d75045d75dd6a8, doi:10.1016/j.ymssp.2017.04.001.

[6] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 2, pages 2051–2058 vol.2, 2001. doi:10.1109/ROBOT.2001.932909.

[7] D. Santosh Kumar. Vision-based robot navigation using an online visual experience, 2007.

[8] B. Crespi, C. Furlanello, and L. Stringa. A memory-based approach to navigation. *Biological Cybernetics*, 69(5):385–393, 1993. URL: http://dx.doi.org/10.1007/BF00199438, doi:10.1007/BF00199438.

[9] R. C. Nelson. Visual homing using an associative memory. *Biological Cybernetics*, 65(4):281–291, 1991. URL: http://dx.doi.org/10.1007/BF00206225, doi:10.1007/BF00206225.

[10] Zhichao Chen and S. T. Birchfield. Qualitative vision-based mobile robot navigation. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2686–2692, May 2006. doi:10.1109/ROBOT.2006.1642107.

[11] Y. Mezouar, A. Remazeilles, P. Gros, and F. Chaumette. Images interpolation for image-based control under large displacement. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 4, pages 3787–3794 vol.4, 2002. doi:10.1109/ROBOT.2002.1014306.

[12] P. De Cristóforis, M. Nitsche, T. Krajník, T. Pire, and M. Mejail. Hybrid vision-based navigation for mobile robots in mixed indoor/outdoor environments. *Pattern Recognition Letters*, 53:118–128, 2015. cited By 0. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84922541097&doi=10.1016%2fj.patrec.2014.10.010&partnerID=40&md5=b4e290d79869080ddbc3269310869e7c, doi:10.1016/j.patrec.2014.10.010.

[13] Last accessed in 2017/5/20. URL: http://www.dictionary.com/browse/contour.

[14] Bill Green. Canny edge detector, 2002. Last accessed in 2017/6/10. URL: http://masters.donntu.org/2010/fknt/chudovskaja/library/article5.htm.

[15] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, February 1962. doi:10.1109/TIT.1962.1057692.

[16] Christoph Dalitz, Christian Brandt, Steffen Goebbels, and David Kolanus. Fourier descriptors for broken shapes. *EURASIP Journal on Advances in Signal Processing*, 2013(1):161, 2013. URL: http://dx.doi.org/10.1186/1687-6180-2013-161, doi:10.1186/1687-6180-2013-161.

[17] OpenCV. Shape matching, May 2017. Last accessed in2017/5/12. URL: http://docs.opencv.org/2.4.13.2/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html#matchshapes.

[18] Last accessed in 2017/5/20. URL: http://dictionary.cambridge.org/pt/dicionario/ingles/template.

[19] OpenCV. Template matching, May 2017. Last accessed in 2017/5/12. URL: http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html.

[20] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, page 147–151, 1988.

[21] Wikipedia. Difference of Gaussians — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Difference%20of%20Gaussians&oldid=747011119, 2017. [Online; accessed 10-June-2017].

[22] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555, Nov 2011. doi:10.1109/ICCV.2011.6126542.

[23] Edward Rosten and Tom Drummond. *Machine Learning for High-Speed Corner Detection*, pages 430–443. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. URL: http://dx.doi.org/10.1007/11744023_34, doi:10.1007/11744023_34.

[24] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary robust independent elementary features. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 778–792, Berlin, Heidelberg, 2010. Springer-Verlag. URL: http://dl.acm.org/citation.cfm?id=1888089.1888148.

[25] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT] or SURF, booktitle = Proceedings of the 2011 International Conference on Computer Vision, series = ICCV '11, year = 2011, isbn = 978-1-4577-1101-5, pages = 2564–2571, numpages = 8, url = http://dx.doi.org/10.1109/ICCV.2011.6126544, doi = 10.1109/ICCV.2011.6126544, acmid = 2356268, publisher = IEEE Computer Society, address = Washington, DC, USA,.

[26] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999. doi:10.1109/ICCV.1999.790410.

[27] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008. URL: http://dx.doi.org/10.1016/j.cviu.2007.09.014, doi:10.1016/j.cviu.2007.09.014.

[28] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 36.1–36.10. BMVA Press, 2002. doi:10.5244/C.16.36.

[29] A. Alahi, R. Ortiz, and P. Vandergheynst. FREAK: Fast retina keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517, June 2012. doi:10.1109/CVPR.2012.6247715.

[30] Arduino Playground. Reading rotary encoders, January 2010. Last accessed in 2016/12/12. URL: http://playground.arduino.cc/Main/RotaryEncoders.

[31] Jean-Yves Bouguet. Matlab camera calibration toolbox, October 2015. Last accessed in 2016/12/12. URL: http://www.vision.caltech.edu/bouguetj/calib_doc/.

[32] Hugin. Hugin - panorama photo stitcher, September 2016. Last accessed in2017/5/12. URL: http://hugin.sourceforge.net/.

[33] OpenCV. Features in opencv, May 2017. Last accessed in 2017/6/2. URL: http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_table_of_contents_feature2d/py_table_of_contents_feature2d.html.